Deliverable D3.8

Impact of Reusable Communication Mechanisms and Hierarchies in Cooperation Strategies and Incentives

Grant Agreement	257513	
Date of Annex I	13-07-2012	
Dissemination Level	Public	
Nature	Report	
Work package	WP3 – Network empowerment	
Due delivery date	28 February 2013	
Actual delivery date	17 June 2013	
Lead beneficiary	UniS	Stylianos Georgoulas, s.georgoulas@surrey.ac.uk

SEVENTH FRAMEWORK PROGRAMME

FUTURE NETWORKS

Authors	ALUD - Siegfried Klein, Markus Gruber FT - Richard Combes, Zwi Altman INRIA - Martin Barrere, Remi Badonnel, Olivier Festor NEC - Johannes Lessmann, Paulo Loureiro TIS - Antonio Manzalini UCL - Marinos Charalambides, Daphne Tuncer UniS - Stylianos Georgoulas, Ning Wang, Xu Zhang, Klaus Moessner UPRC - Panagiotis Demestichas, Evangelia Tzifa, Kostas Tsagkaris, Yiouli Kritikou, Marios Logothetis, Andreas Georgakopoulos, Dimitris Karvounas, Nikos Koutsouris, Asimina Sarli, Aristi Galani, Panagiotis Vlacheas, George Poulios, Vassilis Foteinos, Alexandros Antzoulatos, Petros Morakos VTT - Teemu Rautio, Petteri Mannersalo, Jukka Makela

Executive summary

This deliverable builds upon the work on cooperation strategies and incentives of task T3.4 presented in deliverable D3.4 [1] and presents the updates in the work reported there. The focus of the task is on mechanisms that exhibit strong cooperation aspects. Cooperation can refer to interactions between multiple networking functionalities for achieving a common objective. It can also refer to conditions where multiple, even totally contradicting, objectives of different networking functionalities as well as multiple objectives of different network segments and/or different services need to be taken into account. The granularity of cooperation also varies; it can be contained within the boundaries of a single Network Empowerment Mechanism (NEM) but it can also refer to inter-NEM relationships; in the latter case the mechanism guiding the cooperation between different and independent NEMs is referred to as a UMF Core Mechanism

One notable difference compared to deliverable D3.4 is that in the current deliverable this discrimination between mechanisms exhibiting cooperation aspects and "residing" within a single NEM (intra-NEM) and mechanisms guiding inter-NEM interactions is made clear. While the intra-NEM mechanisms also have the potential to be generalised and adapted for guiding inter-NEM interactions, UMF Core Mechanisms are by definition designed in a way that they are generic and -to a certain extent- NEM agnostic so they can be easily reused in different contexts. It is worth pointing that the UMF core mechanisms mentioned in this deliverable, all belong to the Coordination block (COORD) and have been highlighted in the D2.2 deliverable [2].

It is the role of Chapter 2 to present the project developed NEMs and highlight the cooperation aspects that guide their behaviour. As it will be shown, the developed NEMs vary significantly in the network and service context as well as in their objectives and the methodology they use to implement the required, cooperative in nature, functionality. This functionality -as mentioned above- can refer to meeting multiple objectives of different networks and/or services but also to the condition that the NEMs are distributed in nature, with cooperation needed to guide the interactions between the disjoint NEM components, even if it is a same unique objective they are working towards to.

In Chapter 3 we present the project developed UMF Core Mechanisms. Here, contrary to WP2 - Unified Management Framework, we present the mechanisms not in an abstract and generic way but when applied to specific network scenarios and to specific NEMs. One can consider what is presented here as instantiations of the UMF Core Mechanisms so their applicability and performance can be quantified. Hierarchical optimization, synchronous control theory and centralized multi-objective optimization are considered here, spanning almost the whole range of the project developed UMF (COORD) Core Mechanisms.

Chapter 4 focuses on integration aspects and attempts to present the logic on how the above described NEMs and instantiations of COORD Core Mechanisms should be considered together when needed so by the demands of a networking scenario. Towards this end we first present a set of factors that would need to be considered so that conflicting situations on a per-NEM/mechanism instantiation level can be avoided. As it will be shown there exist multiple factors that need to be considered; here we will present the logic behind them, the actual combination of these factors and their importance/ranking is an on-going activity within WP2.

In conclusion, this deliverable shows that cooperation is a very important aspect both at an intra-NEM level and also at an inter-NEM level and mechanisms to guide this cooperation are absolutely essential so that multiple networking functionalities can work properly together towards a greater goal rather than uncontrollably competing with each other leading to conflicts, instabilities and oscillatory behaviours. The effectiveness of the considered mechanisms is shown through simulations, analytical modelling and proof of concept prototypes. The specific instantiations of COORD Core Mechanisms are an indication of their reusability and examples of successful mapping of mechanisms, from a specification point of view (as in WP2 documents) to an actual scenario/NEM-basis point of view.

Further results with respect to the reusability of the mechanisms are expected in Deliverable D3.9 "Handbook on optimization, learning, operation and cooperation methods" which by its nature will provide a benchmarking of mechanisms; this will provide practical guidelines to guide the actual selection of mechanisms to achieve cooperation both at an intra-NEM but also -and most important- at an inter-NEM level, subject to performance, network and service environment and other constraints.

Table of Content

Foreword **6**
1 Introduction **7**
2 Network Empowerment Mechanisms **9**
2.1 Introduction 9
2.2 Hybrid Peer-to-Peer Selection for Optimized User and Network Performance 10
2.2.1 Context of the work 10
2.2.2 Content of the work 10
2.2.3 Merit of the work 13
2.3 Distributed Decision Engine (Wireless Load Balancing) 14
2.3.1 Context of the work 14
2.3.2 Content of the work 14
2.3.3 Merit of the work 16
2.4 Decentralized and Adaptive Network Resource Management 17
2.4.1 Context of the work 17
2.4.2 Content of the work 17
2.4.3 Merit of the work 20
2.5 Cooperative Remediation of Vulnerabilities 21
2.5.1 Context of the work 21
2.5.2 Content of the work 22
2.5.3 Merit of the work 24
2.6 Balancing of Cell Range Extension and Almost Blank Subframes in 3GPP LTE HetNets (LTE HetNet Optimization) 24
2.6.1 Context of the work 24
2.6.2 Content of the work 25
2.6.3 Merit of the work 26
3 Interacting Network Empowerment Mechanisms **27**
3.1 Introduction 27
3.2 Categories of UMF (COORD) Core Mechanisms 27
3.3 Inter-Cell Interference Coordination (ICIC) and Coverage and Capacity Optimization (CCO) Coordination 28
3.3.1 Context of the work 28
3.3.2 Content of the work 29
3.3.3 Merit of the work 33
3.4 SON Coordination Strategies in LTE-advanced Networks 33
3.4.1 Context of the work 33
3.4.2 Content of the work 34
3.4.3 Merit of the work 39
3.5 Orchestration of Resources and Functions in Edge Networks 39
3.5.1 Context of the work 40
3.5.2 Content of the work 41

3.5.3 Merit of the work 43
3.6 Coordinated Link and Node Load Balancing for Virtualized Evolved Packet Core 44
3.6.1 Context of the work 44
3.6.2 Content of the work 45
3.6.3 Merit of the work 47
4 Interactions: Factors and Considerations 48
4.1 What constitutes a conflict 48
4.2 Behaviour and Specification Factors for Conflict Identification and Resolution 49
4.3 NEMs and Core Mechanism Instantiations: Relationship with Factors 50
4.3.1 Hybrid P2P selection for optimized user and network performance 50
4.3.2 Distributed decision engine 50
4.3.3 Decentralized and adaptive network resource management 50
4.3.4 Cooperative remediation of vulnerabilities 50
4.3.5 Balancing CRE and ABS in 3GPP LTE HetNets 51
4.3.6 ICIC and CCO coordination 51
4.3.7 SON Coordination strategies in LTE-advanced networks 51
4.3.8 Orchestration of resources and functions in edge networks 52
4.3.9 Coordinated link and node load balancing for virtualized evolved packet core 52
4.4 Implications 52
4.4.1 Hybrid P2P selection for optimized user and network performance 53
4.4.2 Distributed decision engine 53
4.4.3 Distributed and adaptive resource management 54
4.4.4 Cooperative remediation of vulnerabilities 54
4.4.5 Balancing CRE and ABS in 3GPP LTE HetNets 54
4.4.6 ICIC and CCO coordination 55
4.4.7 SON coordination strategies in LTE-advanced networks 55
4.4.8 Orchestration of resources and functions in edge networks 55
4.4.9 Coordinated link and node load balancing for virtualized evolved packet core 56
4.5 Discussion 56
5 Conclusion 57
References 58
Abbreviations 61
Definitions 64

Foreword

The main objective of the UniverSelf project is to develop a Unified Management Framework (UMF) that allows for trustworthy interoperability of individual autonomous functionalities. We term the functionality Network Empowerment Mechanism (NEM), which is defined as *"a functional grouping of objective(s), context and method(s) where "method" is a general procedure for solving a problem. A NEM is (a priori) implemented as a piece of software that can be deployed in a network to enhance or simplify its control and management (e.g. take over some operations). An intrinsic capability of a NEM is to be deployable and interoperable in a UMF context (in a UMF-compliant network)."*

In UniverSelf these individual functionalities are approached from three different perspectives in three different work packages (WPs). From the WP3 perspective, the focus is on the algorithmic view (method). From the WP2 perspective, the focus is on the integration view (framework) referring to the capability of a NEM to be deployable in a UMF context (in a UMF-compliant network) considering the interdependencies of the functionalities. From the WP4 perspective, the system view prevails as a NEM is (a priori) implemented as a piece of software that can be tested, validated and deployed. In this context, the ultimate goal of the project is to "put everything together" in a comprehensive and elegant way.

The Network Empowerment work package (WP3) aims to provide the most efficient methods to deliver a toolbox of solutions covering selected operator scenarios. It covers all the work needed to study, design and to evaluate various algorithms with self-x and cognitive capabilities (hereafter termed methods) together with the requirements for their embodiment into network functions to assure trustworthy federation of heterogeneous networks. The work in this work package is based on use cases' problems and should provide the best-suited methods to solve these problems.

The main focus of this deliverable (D3.8) is the work on cooperation strategies and incentives of task T3.4. The context and extent of cooperation can vary; it can refer to interactions between multiple networking functionalities for achieving a common or very similar objectives, but also to conditions where multiple -even very contradicting- objectives of different networking functionalities as well as multiple objectives of different network segments and/or different services need to be taken into account. The granularity of cooperation can also vary; it can be contained within the boundaries of a single Network Empowerment Mechanism (NEM) but it can also refer to inter-NEM relationships; in the latter case the mechanism guiding the cooperation between different and independent NEMs is referred to as a UMF (COORD) Core Mechanism.

This means that the deliverable also links with WP2 in the sense that the defined UMF Core Mechanisms are instantiated here and are bound to specific NEM and networking scenarios so their applicability and performance can be quantified. In all cases, being inline with the DoW, this deliverable presents algorithms/mechanisms "so that actions by individual (potentially selfish) entities translate into desired quasi optimal configurations for a whole ecosystem…modifying the behaviour for improving certain utility functions".

In addition this deliverable also attempts to present the logic on how the described NEMs and instantiations of COORD Core Mechanisms should be considered together when needed so by the demands of a networking scenario, in what can also be considered as a "bridge" between WP3 and WP2. This approach follows the logic behind the instantiations of the COORD Core Mechanisms and while not exhaustive or conclusive, this integration exercise can provide valuable insights on what a UMF-compliant integration should entail and how it should proceed in terms of reasoning.

It is worth noting that the approaches presented here either in the form of NEMs or Core Mechanisms, naturally do parameter optimization and/or exhibit learning in their behaviour. The key difference though compared to approaches presented in deliverable D3.5 - Adaptation and fine tuning of parameter optimization methods [3] and in deliverable D3.6 - Adaptation of learning and operation methods to specific needs of future networks and services [4] is the strong emphasis here in cooperation, in the forms described above.

1 Introduction

The main goal of the Task 3.4 "Cooperation strategies and incentives" is to "identify methods and strategies best-suited for cooperation of entities in the relevant use cases" (DoW). Towards this end, this deliverable presents the outcomes of the project's on-going effort in this direction. In contrast to deliverables D3.5 [3] and D3.6 [4] that are the most recent outcomes of tasks T3.2 and T3.3, the main focus of D3.8 is on the cooperation aspects. That is, while the approaches presented here do parameter optimization and may also exhibit learning in their behaviour, their key characteristic is that they do require cooperation between multiple entities for the needs of optimization and that the optimization itself can have to take into account multiple and contradicting objectives from different network segments and/or services; objectives that are usually seen and dealt with in isolation.

In this direction the main goals of this deliverable are to:

- present the current status of the NEMs developed in the context of task T3.4; that are NEMs that exhibit strong cooperation aspects in their operation either due to their distributed nature or due to the kind of optimization they are attempting, which attempts to balance multiple and contradicting objectives (or both).
- present instantiations of UMF (COORD) Core Mechanisms that guide the interactions between independent NEMs, in a way that otherwise conflicting and uncontrollable behaviours of the NEMs can be streamlined towards a wider objective; in many cases that wider objective takes into account the individual objectives, usually combined/weighted to construct a global objective but it is also possible, as it will be shown, that the individual objectives can be kept as they are and considered still separately but in a way that ensures their parallel optimization.
- present the logic on how the above described NEMs and instantiations of COORD Core Mechanisms should be considered together when needed so by the demands of a networking scenario.

Towards this end, in Chapter 2 we present the corresponding project developed NEMs and highlight the cooperation aspects that guide their behaviour. As it will be shown, cooperation when it comes to individual NEMs' behaviour is something that can prove useful in a diverse set of network and service scenarios and can be implemented in many ways; each time to suit the particularities of the specific context as well as practical constraints in the operation of NEMs. Cooperation is also beneficial not only in the context of optimising NEMs (i.e. NEMs that are meant to be regularly taking decisions and enforcing network and service parameter configurations) but also in the context of knowledge building NEMs; in our particular case a NEM dealing with detection and remediation of distributed vulnerabilities spanning multiple network devices, which is expected therefore to take decisions and enforce configurations only during abnormal events.

In Chapter 3 we present how instantiations of UMF (COORD) Core Mechanisms can be applied to specific network scenarios and to specific NEMs. Examples of hierarchical optimization, synchronous control theory and centralized multi-objective optimization are considered, spanning almost the whole range of the project developed UMF (COORD) Core Mechanisms. By the presentation of the "to be controlled" NEMs one can also reason about the logic behind the adoption of the specific Core Mechanism each time. In order to ease the presentation, a very brief introduction on these Core Mechanisms will also be given and when presenting each instantiation of a Core Mechanism, the category to which it belongs will be explicitly mentioned.

All distinct pieces of work presented in Chapter 2 and 3 follow the same presentation approach, which:

- first briefly sets the context under which they operate (i.e. what is the problem and network/service scenario they address and why),
- then it presents the methodology and algorithmic aspects as well as the obtained results that quantify the performance,
- and, finally, clearly summarizes what the contribution achieved with respect to the problem that it set out to tackle.

Chapter 4 focuses on integration aspects and we present a set of factors that would need to be considered so that conflicting situations on a per-NEM/mechanism instantiation level can be avoided. As it will be shown there exist multiple factors that need to be considered, they are all briefly explained, and the implications they have for the described NEMs and Core Mechanisms instantiations are listed. As these are linked to what can be

considered a “conflicting situation”, what we consider as conflict will also be briefly described so that the logic behind the selection of these factors and their inter-relation can be easily understood.

Finally in the concluding section we summarize the main points and achievements of the work presented in this document, together with directions for future work and improvements; these are to be reported in the D3.9 deliverable.

2 Network Empowerment Mechanisms

2.1 Introduction

In this section, the project developed NEMs, which inherently require the cooperation of many entities or try to balance multiple diverse objectives for their proper operation, are presented.

As aforementioned, cooperation can refer in the simplest case to NEMs being distributed in nature, therefore requiring cooperation and sharing of information between the involved networking entities in order to reach the NEMs' overall objectives. But it can also refer to the fact that a NEM, while centralized, it tries to simultaneously balance conflicting objectives coming from different services, traffic types and/or network segments, requiring cooperation between all these involved notions and the NEM in order to ensure that all these requirements are taken into account. These aspects of cooperation are illustrated in Figure 1.

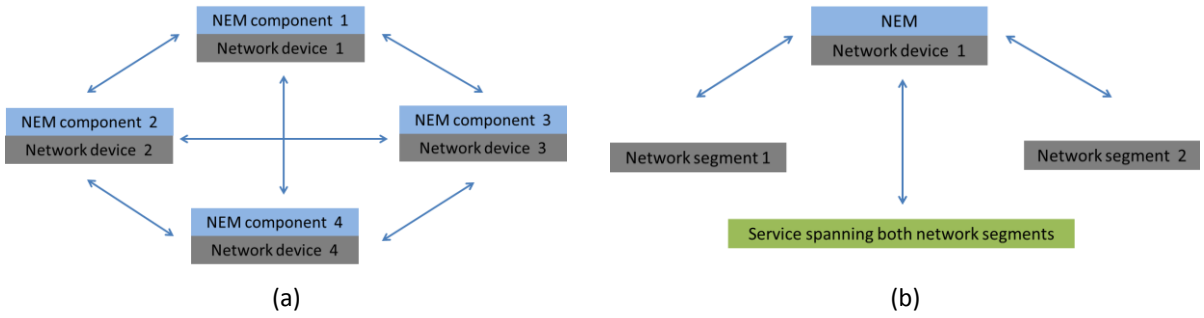


Figure 1. Aspects of cooperation: (a) distributed NEM, (b) cooperative optimization.

In total five diverse NEMs developed in the context of the UniverSelf project are presented, four of which are optimising NEMs meaning that they are meant to always take and enforce actions that attempt to optimize and/or balance certain objectives at core and access network segments so as to maintain application and network performance. The fifth NEM deals with detection of vulnerabilities spanning more than one network devices and their distributed remediation through a collective set of actions performed by multiple devices. This means that while the NEM can be considered as "always ON", actions are actually taken only in cases of events that require remediation. As such, the dynamicity of actions for the latter NEM is expected to be much lower than the other four NEMs, but one would expect that these infrequent actions are to be regarded with higher importance and priority depending also on the operator's priorities.

As such, whenever needed, if these actions lead to changes in the environment and capabilities of the networking devices that the other NEMs manage, these changes and any new constraints should be taken into account by the other NEMs. It is the role of the COORD block in such cases to update the impacted NEM configurations accordingly so that the can operate in the new context.

The NEMs presented in this deliverable are summarized in Table 1 (acronyms will be resolved in the corresponding sections) in order of presentation in the rest of this Chapter.

Table 1. NEMs presented in this deliverable.

NEM name	NEM type
Hybrid P2P Selection	Optimizing NEM
Distributed Decision Engine (Wireless Load Balancing)	Optimizing NEM
Decentralized and Adaptive Network Resource Management	Optimizing NEM
Cooperative Remediation of Vulnerabilities	Knowledge building NEM
LTE HetNet Optimization	Optimizing NEM

2.2 Hybrid Peer-to-Peer Selection for Optimized User and Network Performance

2.2.1 Context of the work

Overlay applications and especially Peer-to-Peer (P2P) applications are generating large volumes of traffic and account for a substantial proportion of the overall internet traffic (according to [5] and [6] P2P flows account for 50%-70% of the overall internet traffic). One notable characteristic of P2P traffic is that it can influence in many cases adversely the network operations and the performance of non-P2P traffic, without though bringing the appropriate revenue to the operator. This is because P2P traffic may have completely different optimization requirements than the operator and as a result it can "bypass" the operator's traffic engineering policies and also react to traffic engineering changes of the operator, in response to what the P2P traffic observes as reduced performance from its side. In addition, the majority of P2P content requests cannot be satisfied locally, therefore the interconnections/relationships with other domains need to be taken into account for the efficient management of P2P traffic.

Certain approaches aim to address these issues by promoting the cooperation between P2P traffic and the underlying network segment and also the cooperation between the underlying network segment with other network domains when possible.

In particular, traditional *locality aware* approaches [5] [7], to which we will be referring to as *non-cooperative strategies*, mainly promote the selection of peers located in the same network, otherwise potential peers in remote domains with the shortest AS-hop distance are selected, but without distinguishing between the inter-domain paths regarding the diversity in business relationships between ISPs. This can lead to revenue loss though for the operator under consideration; for example peers selected in a customer domain can lead to increased revenue since the customer domains are the ones that will have to pay for the flow of traffic across the inter-domain links. However in the non-cooperative strategies, peers from provider domains may be selected instead which will lead to revenue loss since it is the operator under consideration that will have to pay the provider domain for the incurred inter-domain traffic.

A few works proposed recently suggest that ISP business relationships should be taken into account in the peer selection process [8] [9]. These approaches are referred to as the *cooperative strategy*. While they can lead to increased revenue for an operator, they can lead to P2P traffic being routed through a very limited number of inter-domain links and result to severe congestion which can affect the performance of both P2P and non-P2P traffic. In addition it cannot always be guaranteed that peers selected this way will be actually available since domains which may be "losing" in such a scenario may refuse to participate in such a scheme (e.g. by blocking P2P traffic to/from certain domains)

In this work we promote a hybrid peer selection scheme that attempts to bridge the gap between *cooperative* and *non-cooperative* strategies, in order to improve both P2P traffic and overall network performance taking into account revenue considerations and availability of peers in remote domains.

2.2.2 Content of the work

The main concept of the hybrid P2P selection scheme is that for each P2P content request, the cooperative strategy should be considered as long as this does not lead to congestion (in our current work inter-domain links are considered as being more prone to congestion and expensive compared to intra-domain links therefore only the congestion status and cost of inter-domain links is taken into account). Under this strategy, for each P2P content request, peers at the local domain are considered first. If not sufficient peers in the local domain can be found then peers at customer domains are considered. If not sufficient peers in the customer domains can be found then peers at peering domains are considered and, finally, if not sufficient peers in the peering domains can be found then peers at provider domains are considered. This process is inline with the revenue maximization objective.

However, to avoid the congestion issues that it may lead to, whenever a congestion situation is diagnosed as a result of having been following the cooperative strategy, the non-cooperative strategy is followed instead. Under the non-cooperative strategy, in addition to local peers, peers can be selected from customer, peering and provider domains without following the previously mentioned sequence. The scheme is designed so that this "leap" from the cooperative to the non-cooperative strategy happens with increasing probability as the

inter-domain link congestion status worsens. In the current work this probability is given by $\lambda_r = \frac{k'}{k}$ where k' is the number of the congested inter-domain links used by P2P traffic whereas k is the total number of inter-domain links used by the P2P traffic, under the cooperative peering strategy.

The behaviour of the scheme is captured by the following Markov model and a set of differential equations which are summarized in Table 2. In this model, states {L, P, O} correspond to customer ISPs (L), peering ISPs (P) or provider ISPs (O) under the cooperative strategy with (0) being the initial state. And states {L', P', O'} correspond to the states under the non-cooperative strategy, except for the state where the system is in an initial random peering decision making stage (0'). The transition rates between states depend on the inter-domain link congestion status and the mean time for peers participating in transferring desired objects (λ^{-1}) and successfully downloading objects (μ^{-1}); for more details on this the interested reader can refer to [10].

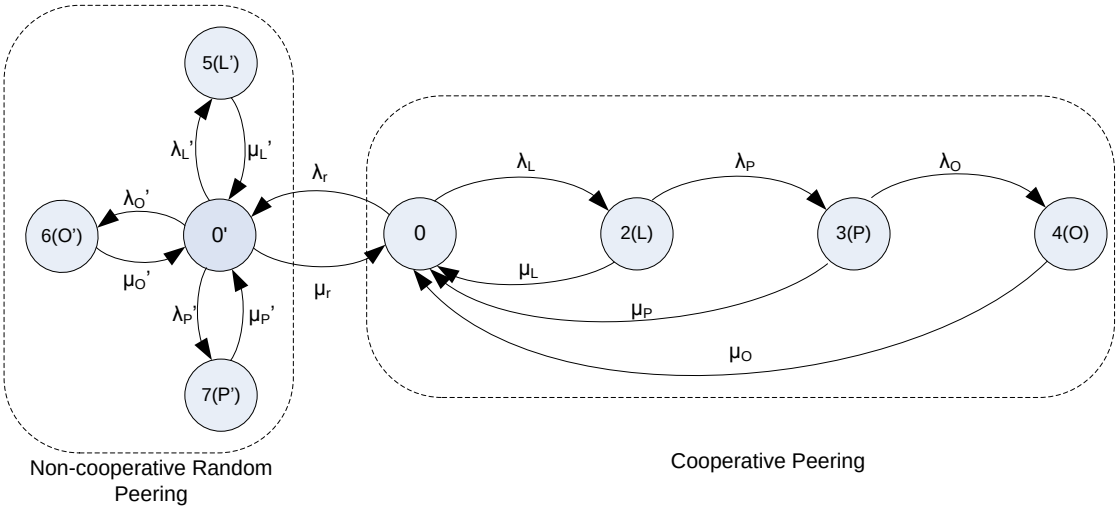


Figure 2. Continuous time markov chain of the hybrid peer selection procedure.

Table 2. Differential equations for the markov model.

$$\frac{dP_0(t)}{dt} = -(\lambda_L + \lambda_r)P_0(t) + \mu_r P_1(t) + \mu_L P_2(t) + \mu_P P_3(t) + \mu_O P_4(t)$$

$$\frac{dP_1(t)}{dt} = \lambda_r P_0(t) + \mu_{L'} P_5(t) + \mu_{O'} P_6(t) + \mu_{P'} P_7(t) - (\mu_r + \lambda_{L'} + \lambda_{O'} + \lambda_{P'})P_1(t)$$

$$\frac{dP_2(t)}{dt} = \lambda_L P_0(t) - (\lambda_P + \mu_L)P_2(t)$$

$$\frac{dP_3(t)}{dt} = \lambda_P P_2(t) - (\lambda_O + \mu_P)P_3(t)$$

$$\frac{dP_4(t)}{dt} = \lambda_O P_3(t) - \mu_O P_4(t)$$

$$\frac{dP_5(t)}{dt} = \lambda_{L'} P_1(t) - \mu_{L'} P_5(t)$$

$$\frac{dP_6(t)}{dt} = \lambda_{O'} P_1(t) - \mu_{O'} P_6(t)$$

$$\frac{dP_7(t)}{dt} = \lambda_{P'} P_1(t) - \mu_{P'} P_7(t)$$

Figure 3 shows the P2P user rate efficiency for the hybrid and the cooperative strategy as a function of the number of available peers in the local domain. It represents the percent of queries that are retrieved successfully for each user and its absolute minimum is equal to 0.3 [10].

As it can be seen, the hybrid P2P selection scheme can lead to an increase in P2P efficiency compared to the cooperative strategy by avoiding inter-domain link congestion situations that can lead to degradation of the P2P performance.

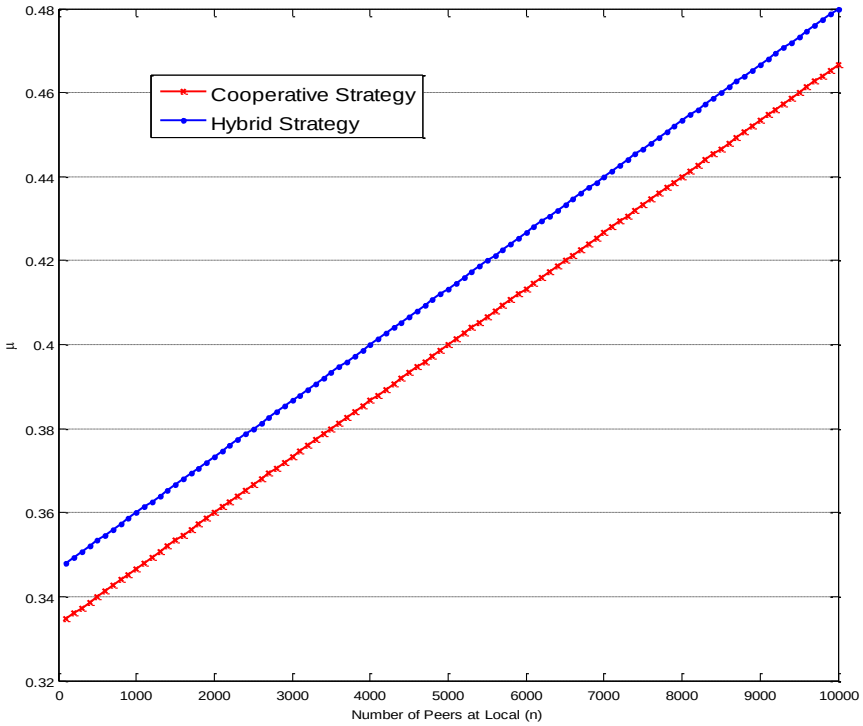


Figure 3. P2P user rate efficiency comparison under different peer selection strategies.

We now investigate the revenue generated by an ISP for carrying P2P traffic. In general, an ISP receives revenue from its subscribers (including customer domains and end users) and pays for the connection to its provider ISPs. The economic cost of an ISP consists of mainly two parts: 1) peering cost, a fixed cost of providing bandwidth from the peering ISP (e.g. for a peering port fee), which is ignored in our work compared to transit cost; and 2) transit costC_j^d, which is a transit cost for each unit of bandwidth to the provider ISP, proportional to the mean allocated bandwidth $\overline{B_d}$. For simplicity we assume that there is an identical charge for both outbound and inbound traffic between a customer ISP and a provider ISP. Therefore, an *ISP*$_i$'s profit can be expressed by:

$$U_{ISP_i} = \left(\sum_{s=1}^{n} c_i I_{(U_s \geq 0)} + \overline{B_d} \sum_{z=1}^{k_c} C_z^d I_{(R_{c\_z} \neq 0)} \right) - \overline{B_d} \sum_{j=1}^{k} C_j^d I_{(R_{o\_j} \neq 0)} \quad (1)$$

where $I_{(\cdot)}$ is the indicator function, and equals to 1 if the condition in the bracket is met and 0 otherwise. Parameter n is the number of local users subscribing to *ISP*$_i$. R_{c_z} and R_{o_j} represent the traffic over links of from customer ISPs to the current ISP and via transit links, respectively, $R_{o\_j} = 0$ means that there is no traffic over transit links connecting its provider domain, and similarly $R_{c\_z} = 0$ means there is not traffic over the links connecting its customer domains (if any). $\sum_s^n c_i I_{(U_s \geq 0)}$ indicates the cost paid by users subscribing to *ISP*$_i$. Term $\sum_{z=1}^{k_c} C_z^d I_{(R_{c\_z} \neq 0)}$ refers to the revenues *ISP*$_i$ generates from its customer ISPs if there are traffic flows over the customer-provider links. $\sum_{j=1}^{k} C_j^d I_{(R_{o\_j} \neq 0)}$ indicates that the *ISP*$_i$ needs to pay transit fees if there are P2P flows exchanged between itself and its *k* multi-homed transit provider ISPs.

Regarding the ISP economic benefits under the hybrid and cooperative peering scenario respectively, since the hybrid peering strategy incorporates random selections of remote domains with the same AS-hops into the peering procedure, this may lead to an uncertain increment involvement of the number of transit links to carry the P2P traffic. Thus the profits for an ISP can be hardly predictable. The reason is that whether the amount of traffic through its customer ISPs is greater than that through its provider ISPs, it is unsure given a fixed number of users at the local domain. That is, for some ISPs, e.g. stub ISPs without customer domains subscribing to, the profits can be impacted much more than those of lower tier ISPs, such as tier-1 ISPs that provide transit connection to customer ISPs for access to the Internet. Therefore, the revenues generated by an ISP can experience decrement under the hybrid peer selection scenario compared to the all cooperation-based peering scenario as shown in Figure 4.

However, if specific requirements can be maintained in the hybrid scenario to transfer the P2P traffic, namely if the relationship between the number of provider-to-customer (k_c), customer-to-provider (k_d) links as a function of the number of available local peers and bandwidth of the inter-ISP links is given by:

$$k_d - k_c \leq \frac{\sum_{i=1}^{n} c_i}{\overline{B_d} C^d} \tag{2}$$

then the profit of an ISP can be guaranteed. It is worth noting though that ISP economic benefits should not be viewed in isolation since they can come at the expense of congestion which can affect the performance of both P2P and non-P2P traffic.

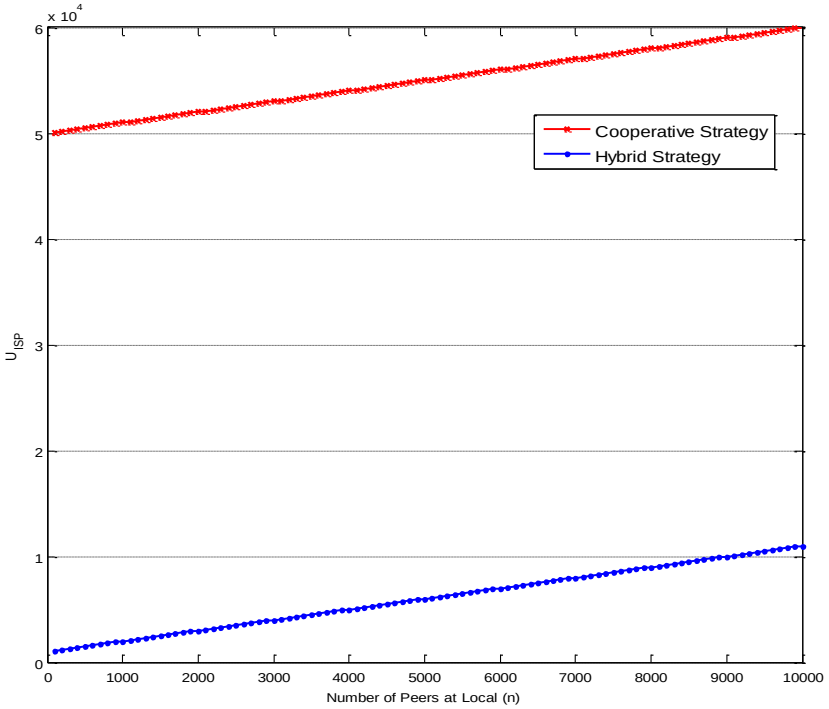


Figure 4. ISP economic benefits comparison under different peer selection strategies.

2.2.3 Merit of the work

This contribution shows that it is possible to combine non-cooperative and cooperative peer selection strategies in a way that can benefit both the P2P traffic and the network operator. This is achieved by "moving" from the cooperative to the non-cooperative strategy whenever there is risk of congestion and also possible failures of the cooperative strategies (e.g. in a networking scenario where inter-domain P2P traffic may be blocked for economic or even privacy issues).

In addition, distributing -when needed- a significant amount of incoming traffic among a potentially higher number of ingress nodes, can ease any traffic engineering NEMs employed by the operator for handling traffic within the boundaries of their core networks when performing their optimizations. This is due to the more

even balancing of traffic among ingress nodes that reduces the chances of overloading certain links inside the domain due to the inherent limitations and constraints of any intra-domain routing scheme.

2.3 Distributed Decision Engine (Wireless Load Balancing)

2.3.1 Context of the work

Cooperative management of shared network resources requires solutions supporting information collection from different sources, distributed processing and information distribution. Specifically, information gathering and provisioning the access selection is one of the more demanding functionalities in the wireless multi-access environment. For example, the existing standard functionalities, such as Access Network Discovery and Selection Function (ANDSF) and Media Independent Handover (MIH), described e.g. in [11] [12], can provide guidance to the selection of the most suitable network, e.g. based on the location, available networks and operator policies. However, the cross-layer and cooperation aspects as well as support for cognitive mechanisms are mostly lacking in these approaches

2.3.2 Content of the work

An enabler for cooperative decision making, called Distributed Decision Engine (DDE) has been developed in the UniverSelf project. Following the similar principles as in [13], DDE is an implemented approach for short time storage and delivery of information (called events in latter) between different entities. Moreover, it provides a means for instantiating (autonomous) network management algorithms and their cooperation.

The architecture of DDE is depicted in Figure 5. It is a collection of software components designed to implement a general distributed publish-subscribe type of message delivery system, which defines a common interface for different information providers (producers), further processing (algorithms) and configuration enforcement entities (consumers) to exchange events. An algorithm is a special case of entity acting as a producer and as a consumer at the same time. EventCache (in the middle of the Figure 5) respectively implements an event caching functionality and is responsible for event distribution based on registrations (a notice of being producer) and subscriptions (a notice of being consumer) kept in the controlling database. Furthermore, an event distribution process, through EventCache(s), can be controlled with access policies, which can be set, for example by operators, to define which consumers are allowed to receive particular event(s). The policies are stored in the policy database. The last functionality of EventCache is a visualization interface that can be used for visualizing the functionality of DDE. In addition, EventCaches can be interconnected in a cascaded fashion, in order to distribute the functionality across multiple nodes in the network.

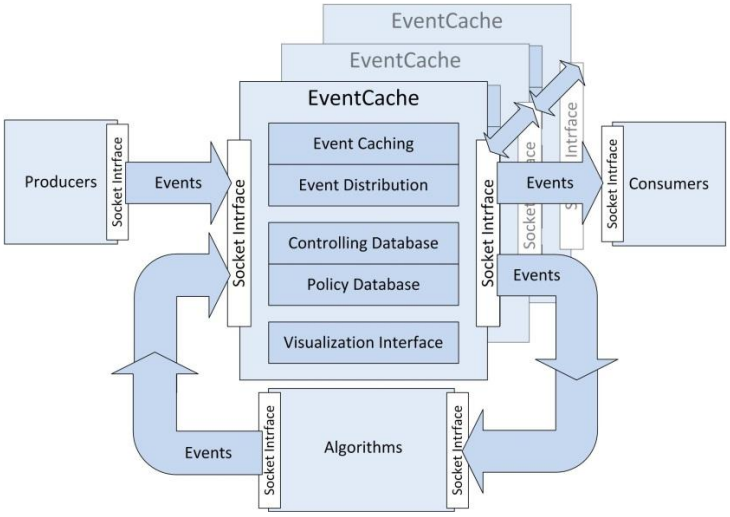


Figure 5. Architecture of DDE.

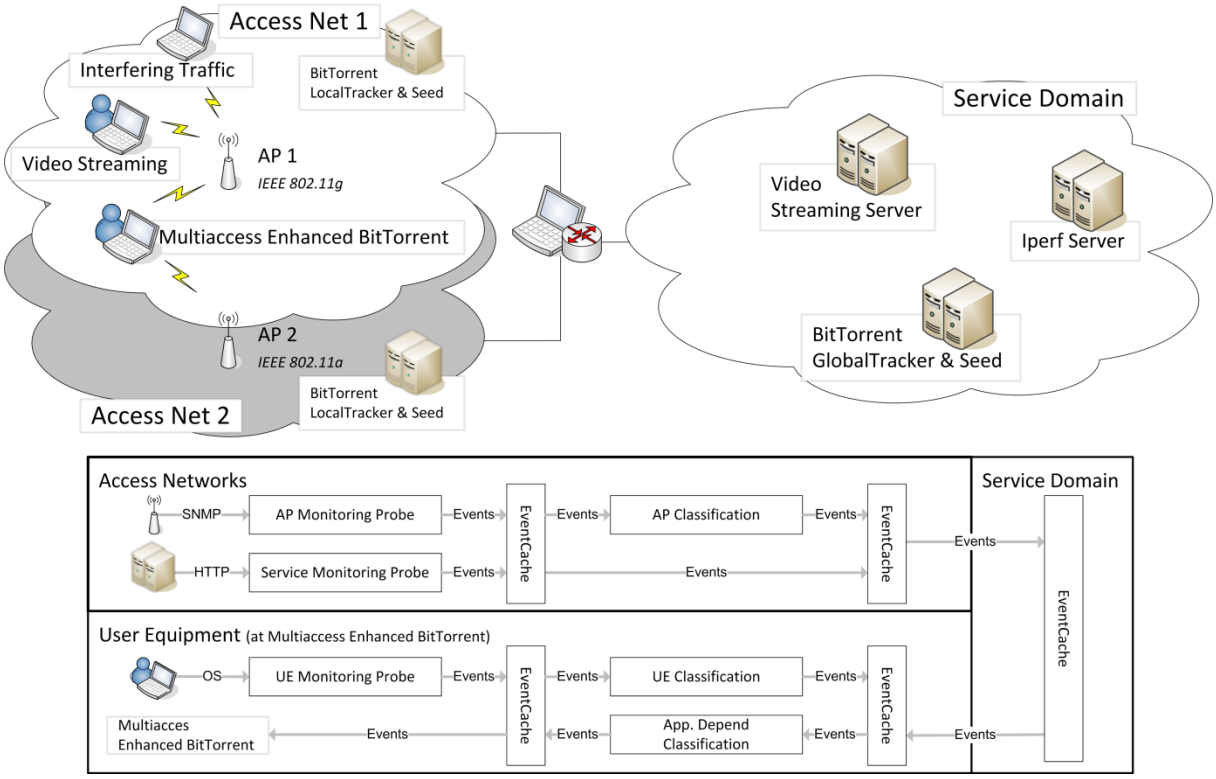


Figure 6. Schematic of the prototype (up) and overall DDE specification for this specific scenario (below).

For communication, DDE specifies five different types of messages: event, registration, subscription, policy and visualization; which all are sent over TCP sockets, and are encoded with External Data Representation (XDR) specified in [14]. Event exchange in DDE relies on the principle that only relevant information is sent over the sockets. In practice, producers are responsible for ensuring that the events to be sent contain significantly updated information, and/or are replacing a soon expiring event as each event is carrying a time to live value. For building a trust between different entities, DDE has a built-in support for self-certifiable event, subscription and policy messages similar manner as [15] does for information.

DDE has been implemented to provide the internal functionalities of the Wireless Load Balancing NEM, which is designed to enable end-to-end QoE-aware resource utilization in a wireless multi-access environment; see [16]. The underlying idea is to select the access(es) for each client so that application/service related information is combined with network and radio link status in the decision process. An overview of the implemented prototype and the corresponding DDE specification are shown in Figure 6. The scenario comprises of two wireless access networks, a router and a service domain, which is connected to the Internet. A multi-access enhanced BitTorrent system [17], including Multi-access Enhanced BitTorrent client, BitTorrent LocalTracker(s) & Seed(s) and BitTorrent GlobalTracker & Seed, is used for demonstrating the advantages of the implemented autonomic resource management mechanisms. The lower part of Figure 6 showcases the main components of the intelligent access selection mechanism. The system collects information from the access points, user devices and services by using the corresponding probes (AP Monitoring Probe, UE Monitoring Probe and Service Monitoring Probe in Figure 6). To support scalable networking solutions, the gathered knowledge is processed in a distributed two-level cascaded way. First the statuses of the access points and devices are determined separately (AP Classification and UE Classification). Then AP and UE statuses are combined together with service/application related information (App. Depend Classification) producing an application-aware classification of the available accesses. Finally, the multi-access enhanced BitTorrent client decides which access(es) to use based on the output of the classification process and current access policies, e.g. set by the operator. In the current implementation, the access grading is based on fuzzy inference but it could be replaced by other classification algorithms too.

A prototype [17] based on DDE was evaluated by a set of measurements and the results proved significant performance improvements in the case of the overloaded access points. For example, Figure 7 shows that

prioritizing video clients and applying traffic offloading to P2P clients, results in good QoE for the video clients and negligible impairments for the P2P clients. On the other hand, Figure 8 shows the signalling traffic measured in the different parts of the prototype during another test run. The AP monitoring probes are sent at fixed intervals, whereas the events to and from the AP classification are triggered only when large enough changes happen. Due to the design of DDE, the overall signalling can be kept reasonable small even in larger networks. This, of course, requires that the management mechanisms utilizing DDE are properly designed.

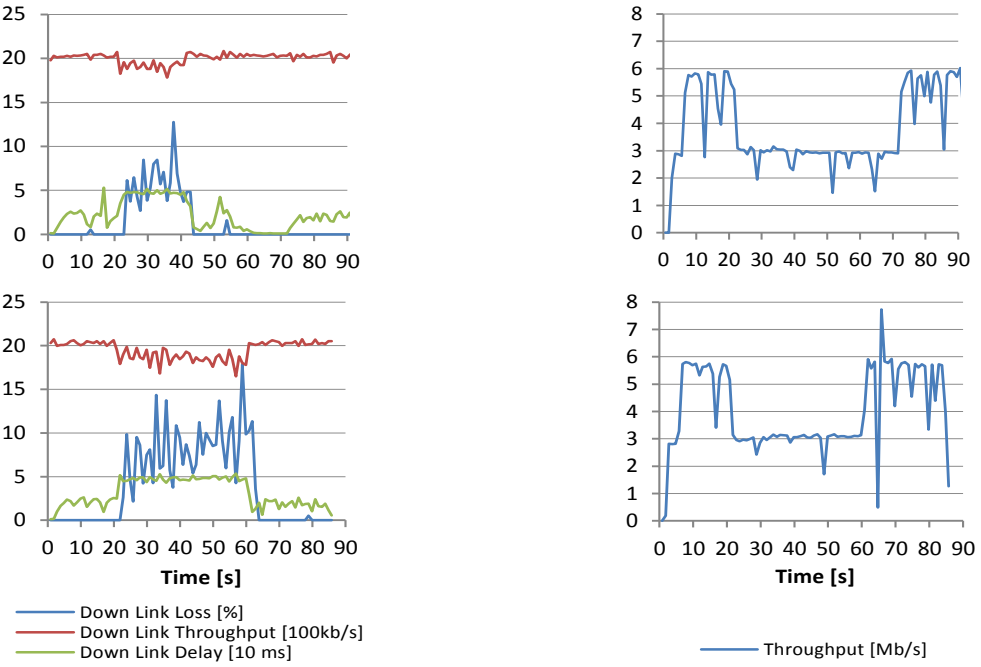


Figure 7. A single test run with DDE enabled prototype (up) and with reference system (below). The access point is congested due to other traffic in the interval [20, 60]. Left, end-to-end QoS statistics of the video client, and right download throughput of multi-access user.

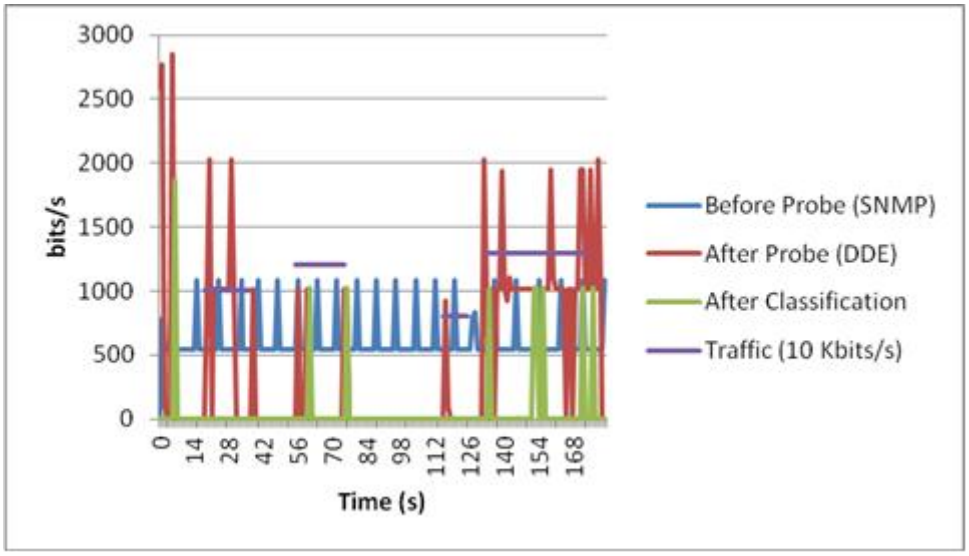


Figure 8. Signalling traffic in different parts of the system during a test run.

2.3.3 Merit of the work

DDE is an implementable framework to test different approaches and solutions within autonomous network management. It provides mechanisms for distributed information collection, event handling, complex self-*

algorithms and cooperation. The developed approach is very flexible and can be easily implemented in different environments. DDE has also been tested in many other scenarios and network environments, including even Internet of Things (IoT) networks.

2.4 Decentralized and Adaptive Network Resource Management

2.4.1 Context of the work

In deliverable D3.4 of the UniverSelf project [1] we had described the main features of a novel adaptive resource management approach for IP networks [18] in which traffic distribution is controlled in an adaptive and decentralised manner according to the network conditions. Based on path diversity provided by multi-topology routing [19], the traffic between any source-destination pair is balanced across several paths according to splitting ratios, which are (re-)computed by the network source nodes themselves. New configurations are not computed by a centralised management entity, but instead, are the result of a real-time adaptation process executed by the source (i.e. ingress) nodes of a network domain. To decide upon the most appropriate course of action when performing periodic re-configurations, the source nodes coordinate among themselves through an in-network management overlay (INO) [20]. This is a logical structure formed by source nodes that facilitates the exchange of information about new configurations.

While in deliverable D3.4 we detailed the functionality of the adaptation algorithm and process, we have extended the approach by investigating two models for organising the source nodes in the INO and by developing a communication protocol to support interactions between INO entities [21]. The performance of the two models is evaluated in terms of re-configuration convergence time and communication overhead.

2.4.2 Content of the work

To achieve the overall objective of balancing the network traffic, re-configuration actions aim to minimise the utilisation of the most loaded link (l_{max}). These actions are taken by individual INO nodes in an iterative manner that dynamically adjusts the splitting ratios of some traffic flows. The node that computes a new configuration at each iteration is known as the Deciding Entity (DE) and its selection is based on a pre-defined rule. In case the DE is not able to determine an acceptable configuration, the decision is delegated to other nodes in the INO, called the Selected Entities (SEs). The interaction between the DE and SEs is supported by the protocol presented below.

Communication protocol

Two different models for the organisation of the INO nodes have been considered. In the first model, all source nodes are logically inter-connected forming a full-mesh topology. In the second model, source nodes are connected according to a ring topology, where each node is connected to only two other INO nodes. This section describes the characteristics of the protocol we have developed, which facilitates the communication between the INO nodes and supports the delegation process in each of the two models.

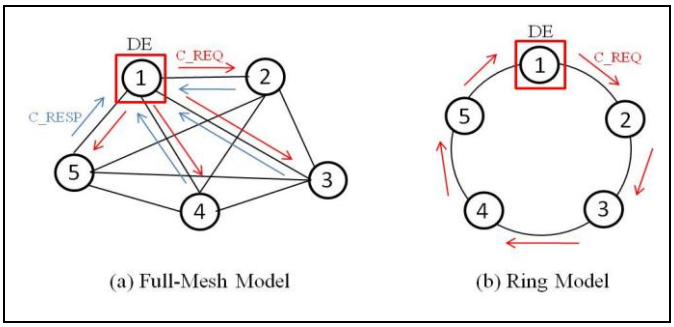


Figure 9. Models to organise source nodes in the INO.

A. Full-mesh model

In this model, INO nodes are connected in a full-mesh topology, as shown in Figure 9(a), where every node can logically communicate with every other node.

To support the delegation process, the developed communication protocol consists of three stages. Upon triggering a delegation process, the DE sends a delegation request -in the form of a COMPUTE_REQUEST (C_REQ) message- to each of its neighbouring nodes (the SEs). The DE then enters a listening period where it waits for replies from all the SEs. Upon receiving a C_REQ message, SEs execute an algorithm to determine if they can perform an acceptable re-configuration according to the conditions described in [21]. The result is copied in a COMPUTE_RESPONSE (C_RESP) message that is sent back to the DE. In addition to compulsory information (such as the success status of any local re-configuration action), the C_RESP message can also include optional information that the DE can use when selecting a solution. This information can be for instance the contribution in terms of volume of traffic of the local flow for which ratio adjustments are proposed to reduce the load of l_{max}. Once the listening period expires, the DE considers all the different C_RESP messages and selects among the successful configurations the one to apply. It then notifies the corresponding SE about its choice by sending an APPLY_REQUEST (A_REQ) message. Upon receiving this message, the chosen SE is responsible for enforcing the re-configuration it had proposed.

Table 3 presents the structure of the messages used in the full-mesh model. Each message consists of a message header and can be extended with optional information elements (IE). Only one IE is typically appended to the messages. According to the action it supports, the message falls into two categories - COMPUTE (driving the execution of the re-configuration algorithm at SEs) or APPLY (driving the choice of the re-configuration decisions to enforce)- that is indicated in the field *Action*. The type of the message (REQUEST or RESPONSE) is indicated in the field *Type*. The result of the re-configuration process is indicated in the field *Status*; the default value is FAIL and is updated by the SEs.

Table 3. Structure of a message in the full-mesh model.

Field	Description
MESSAGE HEADER	
Length	Length of the packet
Action	COMPUTE / APPLY
Type	REQUEST / RESPONSE
Status	SUCCEED / FAIL
Fill bits	Unused bits
OPTIONAL INFORMATION ELEMENT	
ID	Type of appended information
Value	Value of the appended information

B. Ring topology model

In this model, INO nodes are connected according to a ring topology, as shown in Figure 9(b), where each node is connected to only two other nodes. Communication is unidirectional, which means that a node can only pass information to its immediate neighbour in the ring. To communicate with any other node, a message needs to be sent over the ring until it reaches its destination.

The delegation process in the ring model is supported by a two-stage communication protocol as follows. Upon triggering a delegation process, the DE sends a delegation request to only one of its neighbouring nodes (the direction followed can be either clockwise or anticlockwise but this must be fixed). As in the full-mesh model, the request comes in the form of a C_REQ message. The DE then enters a listening period where it waits for the message to travel hop by hop through the ring until it reaches the DE again. Upon receiving the request message, the next hop node analyses the content of the message to decide whether or not to replace the current re-configuration result with its own result. That is, if the contribution in terms of volume of traffic of the corresponding local flow to the load of l_{max} is higher than the one related to the re-configuration currently reported. In that case, the node replaces the current information with the new one and forwards the message to the next hop node. Once the message reaches the DE it is analysed, and, if a successful re-configuration is reported, the DE sends an A_REQ message to the address of the corresponding SE. Upon receiving this message, the SE is responsible for enforcing the re-configuration it had proposed. Compared to the full-mesh

model, where the final selection of a re-configuration action is left to the DE, each node in this model is responsible for determining whether the local solution is more appropriate than the one currently reported. The structure of the messages used in the ring model is similar to the one used in the full-mesh model (see Table 3).

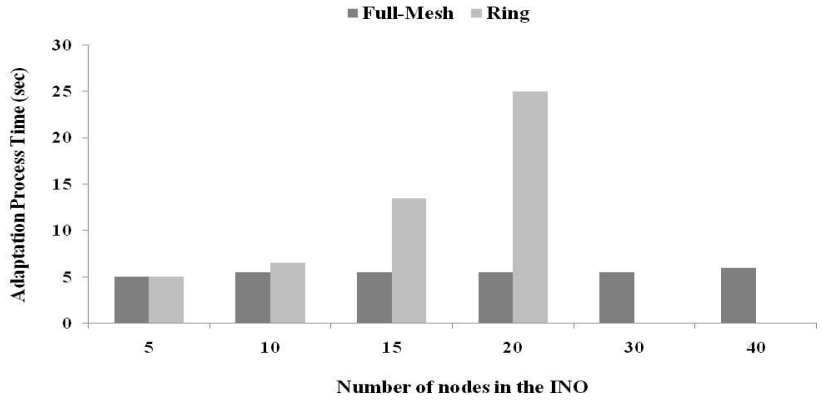
It can be inferred that the waiting time for the DE to obtain the best re-configuration proposal is relatively long, as the message needs to traverse all the INO nodes. In addition, due to the nature of the model, the actual waiting time increases with the number of INO nodes. To avoid this issue in practice, the time required to perform re-configurations needs to be kept small (maximum few seconds) compared to the frequency at which adaptation is invoked (in the order of minutes).

Results

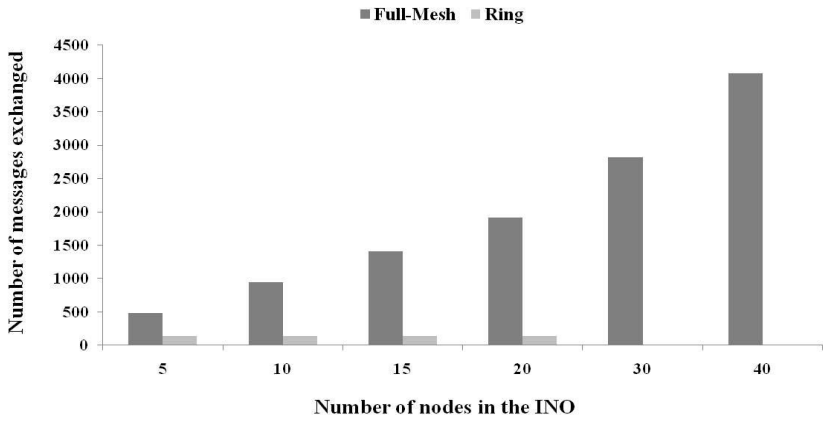
In addition to the performance in terms of resource utilisation gain reported in deliverable D3.4 [1], the overall performance of the proposed resource management approach also relies on the convergence time and cost (in terms of management overhead). Different factors may influence the time required to complete the adaptation process, such as the physical characteristics of the network and the execution of the delegation process at different iterations of the adaptation cycle. The actual time to execute one iteration depends on the execution time of the re-configuration algorithm. In particular, in the best case where no delegation is required, the execution time of the algorithm can be as low as 7ms. In case of delegation, however, the total execution time of the algorithm can be significantly longer given that interaction between physically distant entities is required. Several factors can influence the execution time of this process, such as the structure of the INO, the number of neighbouring nodes, the physical distance between INO nodes, but also, the characteristics of the communication protocol to support the interactions. We analyse here how the two proposed models to organise the INO nodes may affect the performance of the adaptation process, both in terms of convergence time and in terms of overhead associated with coordination among the nodes.

In order to evaluate these factors, we consider a set of nodes that we connect according to the two models, i.e. full-mesh and ring structures. We perform several sets of experiments by varying the number of nodes in the INO and the connectivity model of the nodes. An experimental set involves the emulation of the adaptation process. A node in the INO is randomly selected to be the DE. The adaptation is performed over 50 re-configuration iterations and at each iteration the delegation process is triggered by the DE, which initiates communication with its neighbours. Although delegation may not be triggered at each iteration in a realistic setting, our evaluation considers the worst case scenario. For each set of experiments we investigate the total time required to complete a cycle of the adaptation process (T_{adapt}), i.e. to compute and enforce new configurations, and we determine the volume of coordination messages required during the adaptation. As it will be shown, 50 iterations provide a good balance between adaptation process time and configuration accuracy.

Figure 10(a) shows the evolution of T_{adapt} according to the number of nodes in the INO for the two models. We can observe that the total time is not affected by the number of nodes in the full-mesh model, whereas this substantially grows as the number of nodes increases in the ring model. The results also show that the full-mesh model performs better than the ring model in terms of execution time. In fact, the ring model performs as well as the full-mesh for a small number of nodes (up to 10) but shows poor performance with a large number of nodes. Given the poor scalability performance achieved from only 20 nodes in this model, we do not extend the experiments to a larger number of nodes. Even if the actual time required for enabling communication between the different entities may be affected by the physical distance between source nodes, as reported in [22], the results show that the total time required for the adaptation can be kept to an insignificant level (few seconds) compared to the frequency at which the adaptive resource management scheme is invoked (every 10-15 minutes).



(a)



(b)

Figure 10. (a) Evolution of the total execution time, (b) evolution of the total number of coordination messages exchanged.

The evolution of the total number of coordination messages exchanged during the adaptation process is presented in Figure 10(b). We can observe that the actual gap between the number of exchanged messages in the two models increases significantly as the number of nodes in the INO increases. These results show that the ring model scales better than the full-mesh one in terms of communication overhead. Although the number of messages is independent of the number of INO nodes in the ring model, it linearly increases with the number of INO nodes in the mesh approach. Given the small size of coordination messages (typically less than 10 bytes) the overhead incurred by the delegation process is not significant given today's network capacities (a model that improves both the ring and the full-mesh model in terms of latency and message overhead is currently being investigated).

2.4.3 Merit of the work

Current practices for managing network resources rely mainly on off-line traffic engineering (TE) approaches where the expected demand is calculated from previous usage and a specific routing configuration is produced. Given their static nature, these off-line approaches can be well sub-optimal in the face of changing or unpredicted traffic demand. In the proposed scheme, traffic distribution is controlled in an adaptive manner, according to network conditions, and can efficiently deal with network and traffic dynamics by performing adaptations of routing configurations in short time scales (i.e. every 10-15mins). As a result, better and more efficient utilisation of network resources can be achieved.

Furthermore, most on-line TE approaches, e.g. [23] [24], rely on centralised managers that periodically compute new configurations according to dynamic traffic behaviour. These approaches have limitations especially in terms of scalability (i.e. communication overhead between the central manager and devices at run-time) and lag in the central manager reactions that may result in sub-optimal network performance. The decentralised nature of the proposed resource management approach circumvents issues associated with centralisation. The analysis and experimental evaluation of the proposed mechanisms presented both here and in deliverable D3.4, indicate that our approach can achieve near-optimal performance in terms of resource utilisation in only few seconds, without overloading the network with excessive coordination messages.

2.5 Cooperative Remediation of Vulnerabilities

2.5.1 Context of the work

Autonomic environments must support their own management; this includes the capability of detecting and remediating configuration vulnerabilities. We have already shown in [25] that a configuration vulnerability can be spread over several devices in a network. The scenario presented below in Figure 11 illustrates a typical example where two devices, a SIP (Session Initial Protocol) server with no flooding protection and a local DNS (Domain Name System) server with external unknown name resolution, constitute a distributed vulnerable state. In this situation, an attacker can perform a denial of service attack by flooding the SIP server with unresolvable domains that must be solved by a local DNS server. The local DNS server in turn is configured for solving unknown domains querying external servers, thus increasing the number of waiting requests as well as the response time for each SIP request. If at least one of the servers is not present or is not compliant with the required specific state, then the distributed vulnerability does not exist. In order to correct such security problem, different remediation tasks could be performed in the SIP server or in the DNS server. When configuration vulnerabilities have been identified in a network, it is important to determine a proper strategy for determining how and by which devices the distributed vulnerability can be remediated.

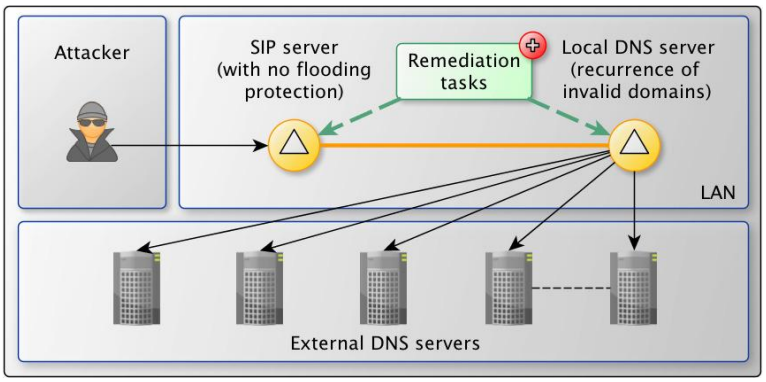


Figure 11. Remediation of distributed vulnerabilities.

Commonly, vulnerabilities take the form of software flaws or misconfiguration errors and they can be usually corrected by means of different methods such as applying software patches, adjusting configuration settings or removing the affected software. However, when corrective actions are performed changes are introduced in the environment thus change management mechanisms such as those proposed in [26] must be taken into account. Risk assessment methods are also important as they provide a strong basis for analysing the impact of remediation activities within the vulnerability treatment process [27]. It is crucial to ensure safe changes not only from an operational viewpoint but from a security perspective too. While several works have been focused on vulnerability management such as [28], just a few works address this topic on remediation into autonomic environments. Several languages have been proposed for supporting the vulnerability management process. De facto standards such as OVAL (*Open Vulnerability and Assessment Language*) [29] and XCCDF (*eXtensible Configuration Checklist Description Format*) [30] provide means for dealing with device-based vulnerabilities. OVAL is an XML-based language supported by the MITRE Corporation intended to standardize how to assess and report upon the machine state of computer systems, with a particular focus on vulnerability descriptions. The XCCDF language maintained by NIST provides standard means for specifying security

checklists under both technical and non-technical perspectives. In order to also cover distributed scenarios, we have proposed Distributed OVAL (DOVAL) [25], a language built on top of OVAL for describing and assessing distributed vulnerabilities. However, there is a lack for supporting cooperative treatments capable of mitigating and remediating distributed vulnerabilities. In this work, we have focused on the modelling of distributed treatments and their specification (distributed XCCDF (DXXCDF) language defined on top of the XCCDF language), as well as on a cooperative strategy to solve them, where a device can support the security of another one in order to contribute to the overall network security.

2.5.2 Content of the work

In order to design our approach, we have formalized distributed vulnerability treatments by extending and enhancing our previous mathematical model [25]. A distributed vulnerability is defined as a set of conditions over two or more network devices that if observed simultaneously, a potential threat is present on that network. It is important to remark that the required conditions to be observed over a specific device do not necessarily constitute a complete device-based vulnerability description.

Based on our modelling, we consider a distributed treatment (DT) as a body of tasks performed over a set of network devices that introduces configuration changes in order to eliminate the security weakness described by a specific distributed vulnerability (DV) [31]. In order to formally define what a distributed treatment is, we define the following domains: A = {a1, a2,…} denotes the set of actions applicable over network devices, and T = {t1, t2,…} denotes the set of tasks applicable over network devices. A task ti is a logical combination of actions and its logical value is computed based on the successful application of each action. The set T is inductively defined as follows: if ai ∈A, then ai ∈T (i∈N), and if $\alpha, \beta \in T$, then $(\alpha \diamond \beta) \in T$, $\diamond \in \{\wedge, \vee\}$.

Let us consider H = {h1,h2,...} as the set of devices in the network (e.g. hosts, routers), S = {s1, s2,...} the set of device states where a state si describes a set of properties to be observed (e.g. version number of a library), and R = {r1, r2,...} the set of relationships between network devices (e.g. reachability or service provisioning). We have then specified the following set of *core functions* in order to support the application of remediation tasks over the network*:*

- stateH: H → S ≡ function that takes a device h∈H as input and returns its current state s ∈ S,
- stateR: R → 2S ≡ function that takes a network relationship r ∈ R as input and returns a set with the current state si ∈ S of each involved network device hi ∈ H in the relationship,
- action: H × A → H ≡ function that takes a device h ∈ H as input and returns the same device h after performing an action a ∈ A.
- taskH: H × T → H ≡ function that takes a device h ∈ H as input and returns the same device h after performing a task t ∈ T that produces an observable change on its state. This means that at least one action ai ∈ A must introduce a change that cannot be rolled back by any other action in the task nor a combination of them. The following property holds in the considered model: stateH(h) ≠ stateH (taskH (h, t)), ∀t ∈ T, ∀h ∈ H.
- taskR: R × T → R ≡ function that takes a network relationship r ∈ R as input and returns the same network relationship r after performing a task t ∈ T over its member devices. Based on the definition of T, is can be noticed that the task t will produce an observable change on its state and that the following property also holds: stateR(r) ≠stateR(taskR(r, t)), ∀t ∈ T , ∀r ∈ R.
- T^H = {tH1,...,THn} denotes the body of available tasks for performing over network devices where each task tHi is semantically related to a specific state si. Usually, the following equation can hold $|T^H| < |P^H|$, meaning that treatment tasks are not always available for correcting certain device states.
- TR = {tR1,..., tRv} denotes the body of available tasks for performing over network relationships where each task tRi is semantically related to a specific relationship ri. Usually, the following equation can hold $|T^R| < |P^R|$, meaning that treatment tasks are not always available for correcting certain network relationships.

We therefore define a distributed treatment DT as the compliant application of (T^H, T^R) over the network (H, R) that eliminates every possible matching projection of the pattern (P^H, P^R) over (H, R). Under a logical perspective, this is defined as the disjunction of task applications over each potential combination of devices and network relationships (H', R') performing the roles required by the distributed vulnerability DV as follows:

- DT(H,R) = Π(TH,TR) = taskH(h1,tH1) ∨ ... ∨ taskH(hn,tHn) ∨ taskR(r1,tR1) ∨ ... ∨ taskR(rv,tRv)
- ∀H' = {h1,...,hn} ⊆ H,R' = {r1,...,rv} ⊆ R such that DV(H',R')holds.

Changes done for correcting different instances (H', R') of the distributed vulnerability must not shadow performed remediations for other observed vulnerable instances of DV, thus ¬DV (H, R) must hold after the DT application. In order to capture the previous mathematical constructions, we have conceived the DXCCDF language, built on top of XCCDF, as a means for expressing vulnerability treatments in a machine-readable manner. XCCDF rules allow specifying remediation information that can be used by automated systems to perform corrective actions when specific states are detected. These states can be specified by languages such as OVAL and DOVAL. DXCCDF extends XCCDF by considering a new building block named complex-Rule under the dxccdf namespace. This extension provides the ability to specify a Boolean expression involving all the potential tasks that can be performed for remediating a specific machine state. Our cooperative strategy consists in building a spanning tree over the network, where each potential node involved the vulnerability will inform on how it can cooperate in the corrective tasks and at what cost. In that context, we have specified the supporting algorithm in [31]: a spanning tree is first built in order to explore and gather devices information. Each active node of the tree reads the list of roles involved in the vulnerability instances given in the DOVAL document dv, identifies itself in the list, and for each task found in the DXCCDF document applicable on each specific role that the node can play, the task cost is computed by the node itself and attached to the general cost table. The traversal continues on the left and right sub-trees until the whole spanning tree has been explored. This strategy can be performed in parallel to the assessment if treatments are already available at this stage. We have evaluated the performance of our approach in an analytical manner. In particular, Figure 12 shows the time required for performing the assessment and remediation of vulnerabilities while varying the number of devices in the network.

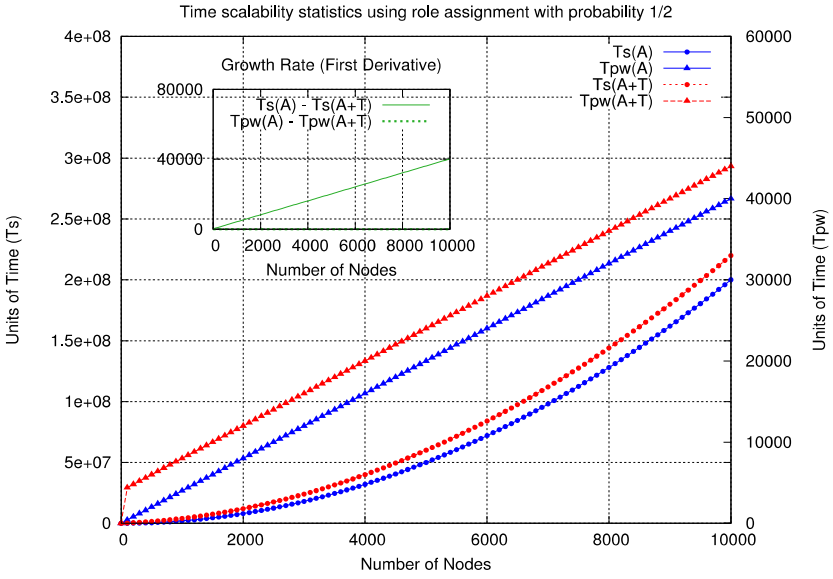


Figure 12. Performance analysis of cooperative remediation.

Both solid blue lines with rounded and triangular points represent the time required when only the assessment process is performed under a sequential (TS(A)) and a parallel (TPw(A)) approach. Dashed red lines show the time behaviour when assessment and remediation activities are performed at the same time (A+T). TS(A+T) illustrated by the dashed red line with rounded points shows the same growth rate than TS(A). The same

phenomenon can be observed when a parallel approach is taken as depicted by TPw(A) and TPw(A+T). First derivatives drawn in the inner graph confirm the same growth rates for both sequential (green solid line) and parallel (green dashed line) approaches, considering only the assessment (A), and both the assessment plus treatment (A+T) activities.

2.5.3 Merit of the work

This contribution shows that it is possible to define a cooperative strategy for supporting the remediation of configuration vulnerabilities. We have first described how distributed treatments with respect to a configuration vulnerability (OVAL definition) can be mathematically modelled and specified based on an extension of the XCCDF language. We have then evaluated how these descriptions of treatments can be exploited in order to identify and select network devices involved in the vulnerability remediation. Each device expresses its capabilities by associating costs to the remediation tasks specified in the description. This approach enables the devices to contribute to the vulnerability solving in a cooperative manner. In particular, a device is not necessarily capable of solving a local vulnerability (due to the lack of patches for instance). In that case, another device (for instance, a firewall) can contribute to the security of this device in favour of the overall security of the network.

2.6 Balancing of Cell Range Extension and Almost Blank Subframes in 3GPP LTE HetNets (LTE HetNet Optimization)

2.6.1 Context of the work

LTE Heterogeneous Network (HetNet) deployments are an answer to the ever increasing demand for higher user data rates. In HetNets, the usual macro cell deployment is complemented by pico cells, which are usually positioned without any network planning into areas with high user densities as shopping malls or urban areas. As radio bandwidth is a scarce resource, pico and macro cells are usually operating in the same spectrum. Because of the much higher transmission power of the macro cells (typically 40W) in comparison to the pico cells (typically 1W), the pico cells need a kind of resource protection for efficient operation.

Different approaches are available for resource protection of pico cells, which typically involve interference coordination between macro and pico cells. Carrier Aggregation, a mechanism foreseen by 3GPP can be employed to coordinate interference in the frequency domain. Another approach described by Ericsson -Soft Cells [32]- foresees common control signalling between macro and pico cells and thereby effectively reduces pollution of the physical downlink control channel (PDCCH) and also allows efficient interference coordination in both frequency and time. A further mechanism defined by 3GPP -Almost Blank Subframe (ABS)- allows efficient interference coordination in the time domain.

The basic transmission interval in LTE lasts 1ms and is called a subframe. The main content of an LTE subframe is reference signals which allow channel estimation, signalling information about transmissions scheduled in a subframe and data transmissions that were announced in the signalling part. Almost blank subframes don't hold neither signalling nor data. Pico and macro are kept synchronous and are exchanging information about ABS with the help of a bit field of length 40. As ABS is relevant to the channel state reporting, each UE has to be reconfigured if the ABS definition is changed within a macro/pico cluster. More details on ABS related signalling are provided by 3GPP in [33] [34].

Compared to the limited range of pico cells, they control an enormous amount of radio resources. This leads to very high data rates for users attached to a pico cell and introduces a high grade of unfairness into the HetNet. One way to cope with this unfairness is to increase the number of users which associate to pico cells by extending their cell range, which is commonly named Cell Range Extension (CRE).

User Equipment (UE) will typically associate to the radio cell providing the highest long term signal. The handover process will continuously monitor and will reassociate a mobile whenever necessary. The handover process includes the cell individual offset parameter which is added to the signal of neighbour cells and which can be negotiated individually between these. In CRE this parameter is named BIAS and it makes use of it to virtually increase the footprint of pico cells. A change of the BIAS also requires the UE handover measurements to be reconfigured. In contrast to CRE reconfiguration this is less expensive, as only the UE located in the

border between the respective neighbour cells are concerned and further this reconfiguration can be done on occasion, e.g. while executing a handover. More details on CRE based signalling is provided in [33] [34].

All cells of an LTE network exchange load information over the so called X2 interface. This load information comprises the average amount of physical transmission resources in ABS and non-ABS subframes in the reporting interval. Details are given in [34].

2.6.2 Content of the work

In this work we focus on the question how to balance the ABS scheme with the CRE under consideration of the restrictions imposed by the 3GPP defined signalling and present a corresponding algorithm which automatically adapts to different environmental conditions. Further all macro cells will apply the same ABS definition.

If a given UE can be served by either a pico or a macro cell, the interference level in the HetNet will be lower if the UE is served by the pico cell. This is due to the lower transmission power of pico cells in comparison to macro cells. In the algorithm design we exploit this general observation by assigning as many UEs as possible to pico cells by increasing the BIAS to the maximum feasible value. Further pico cells dispose of a mechanism to defend against cell overload. This defence mechanism will stepwise decrease the BIAS value with the surrounding macro cells if the average load in a measurement interval is above a predefined threshold. If the overload situation of the pico cell is cleared, i.e. the average load drops below a lower threshold, then the BIAS will be stepwise increased up to the predefined maximum. As the pico cells will typically operate at a high BIAS, the macro cells are obliged to provide ABS. The exact amount of ABS depends on many factors as the user density in hotspots and their size, the number of pico cells per macro cell or the offered load. In general, the macro cell will make use of non-ABS to serve UEs outside the range of Pico cells.

A strategy based on the load of ABS in the pico cells to control the amount of ABS is detailed here. For that purpose the load of ABS resources of all individual pico cells within a measurement interval is combined into a Cumulative Distribution Function (CDF). The 90%-tile of the named CDF represents the load in the pico cells and ensures that only a low number of pico cells face a higher load than the value expressed by this quantile. These cells are able to defend against overload by reducing their BIAS as it was explained above. If the pico cell load exceeds an upper threshold, then the relative amount of ABS in all macro cells is increased by one step; if it falls below a lower threshold, then the relative amount of ABS is decreased. Practically the combination of the pico and the macro resource control algorithms results in a hierarchical structure where the pico cells are employed by the macro cells to offload as much traffic as possible and where the macro cells are providing enough resources to the pico cells to cope with the offered traffic.

This work is assessed by means of system simulation in a downlink scenario widely compliant with the simulation model detailed in [35]. The employed channel model limits to pathloss and position correlated large scale fading in non-line of sight conditions. FTP traffic model 1 according to [35] with a file size of 0.5 MB has been chosen for input traffic in combination with a clustered user dropping according to scenario 4a in the same document. The signal of the individual cells is calculated as function of the cell transmit-power and antenna pattern, the UE antenna model and the radio channel model. After a UE is associated to a serving cell, the signal to interference plus noise ratio (SINR) can be calculated. Caused by the FTP traffic model chosen for this simulation study, radio cells may be idle because there are no data waiting for transmission in a cell. In this case, the corresponding radio cell doesn't transmit data and is therefore not considered as interference in the SINR calculation. The resulting SINR (in dB) is mapped to a channel capacity by the definition below:

$$C(x) = \begin{cases} 0 & x < -7.04 \\ C(20.2) & x > 20.2 \\ -0.0001x^3 + 0.0074x^2 + 0.1397x + 0.6218 & else \end{cases} \quad (3)$$

To calculate the amount of data transferred during a transmission, the channel capacity is multiplied with the bandwidth allocated by the scheduler.

Due to the cell range extension of the pico cells, ABS protected resources are required to schedule UEs located in the cell border. The scheduler of pico cells uses a SINR threshold of 3dB to decide whether to schedule UEs into either ABS or non-ABS resources. The scheduler of the macro cells has to enforce the ABS and is therefore limited to the usage of non-ABS resources.

Table 4 concludes the most important simulation parameters with an emphasis on cell layout, radio channels and configuration of algorithms for control of BIAS and ABS. The large scale fading is spatially correlated with a correlation coefficient of 0.5 at the distance given in parenthesis. The general averaging period for measurements is 1sec. An environmental parameter defining the user density in the hotspots is the hotspot probability P(HS). If a new UE arrives in the simulation, then it is either placed with the given probability into a random hotspot or it is independent identically positioned into the simulation playground.

Table 4. Simulation parameters.

Parameter	Value	Parameter	Value
Macro cell layout	3-sector hexagonal with 500m inter-site distance	Pico cell layout	Randomly placed
Pathloss (Macro)	128.1 + 37.6 log(d)	Large Scale Fading (Macro)	8dB, (0.5 @ 50m)
Pathloss (Pico)	140.1 + 36.7 log(d)	Large Scale Fading (Pico)	10dB, (0.5 @ 25m)
BIAS allowed range	0 – 16 dB	ABS allowed range	0.05 – 0.5
Threshold upper (Pico)	0.8	Threshold upper (ABS)	0.95
Threshold lower (Pico)	0.5	Threshold lower (ABS)	0.6
Step BIAS	1dB	Step ABS	0.025
Load Averaging Time	1sec	UE drop time	8sec

Simulation experiments are executed for different number of pico cells per macro cell and with different user densities in hotspots with the aim to find the offered traffic at which 5% of all arriving users are dropped. Therefore the offered traffic is increased in steps of 4Mbps until the user drop ratio is above 5%. From the simulation experiments with a drop ratio closest to 5%, the corresponding input traffic is interpolated.

The simulation results are concluded in Table 5. Besides the input parameters, the offered traffic at which 5% of all users are dropped is shown. If available, we compare these values to simulation results gained from a static configuration of BIAS (up to 16dB) and ABS (up to 70%) which are shown in parenthesis.

Table 5. Simulation results.

Pico / Macro	P(HS)	Offered traffic (5% dropped)
1	0	12.1 Mbps (12.9 Mbps)
1	1/15	12.3 Mbps (14.2 Mbps)
1	2/15	12.4 Mbps
2	0	12.2 (15.6 Mbps)
2	2/15	13.6 (18.2 Mbps)
2	4/15	15.0 Mbps
4	0	15.1 Mbps (21.7 Mbps)
4	4/15	16.4 Mbps (26.0 Mbps)
4	8/15	20.3 Mbps

2.6.3 Merit of the work

When compared to static settings of ABS and BIAS the proposed algorithm shows good performance with a low number of deployed pico cells per macro cell. At a higher number of deployed pico cells, the algorithm performance degrades significantly. As this algorithm automatically adapts to dynamic environmental conditions it can be employed in a wide range of scenarios without the need to manually fine tune ABS and BIAS settings. Further work is required to optimize the performance of the presented algorithm and to assess the signalling overhead caused by the dynamic reconfiguration of ABS and BIAS in the radio access network.

3 Interacting Network Empowerment Mechanisms

3.1 Introduction

In this section we present approaches that can be used to guide the joint operation of NEMs. That is, while each NEM is an autonomic loop, these approaches operate "one level" higher allowing the streamlining of NEMs' operations towards the objective of broader stability and optimization. These approaches attempt to avoid conflicts due to the concurrent operation of NEMs and also jointly optimize NEMs, whenever there are possibilities of such wider-scale optimizations (the definition of conflict will be introduced in more detail in Section 4.1).

These approaches which fall under the category of UMF (COORD) Core Mechanisms are not exclusively bound to the specific NEMs as presented below, but can be generalised to guide the operation of other NEMs. In principle what is being presented below is instantiations of the mechanisms presented in Section 4.3.1 of the D2.2 UniverSelf project deliverable [2], for specific NEMs and specific networking contexts.

Prior to presenting these instantiations we will very briefly introduce the categories of UMF (COORD) Core Mechanisms so that the mapping of the instantiations on specific types of mechanisms can be easily derived and understood.

3.2 Categories of UMF (COORD) Core Mechanisms

So far, in [2] the following categories of core mechanisms for managing conflicts and concurrent operations of NEMs have been defined:

- hierarchical optimization,
- synchronous control theory (and its asynchronous generalization),
- separation in time strategies,
- and centralized multi-objective optimization

Hierarchical optimization deals with NEMs that operate at different time scales; that is they optimize some network parameters with different frequency. The idea here is to set "slower" NEMs as "leaders" and the "faster" NEMs as "followers" and have NEMs optimize their parameters independently and the faster NEMs seeing the configurations enforced by the slower NEMs as "semi-static" [36]. All NEMs in this approach keep their individual objectives intact and they can operate in parallel at different time scales though.

Synchronous control theory approaches deal with NEMs (processes/functionalities in general) that operate at the same time scale and are synchronized. The idea here is to have NEMs jointly optimize towards their individual objectives, but taking into account the influence of the other NEMs. This can be done by considering a global/aggregated expression of their utilities (optimization objectives) or an expression of the weighted deviation from some pre-defined per-NEM optimization targets. The same approach can also be generalized to be applied for NEMs that are not necessarily synchronized in time but have the same average frequency of parameter changes [37].

Separation in time strategies [2] on purpose try to have only one NEM enforcing a parameter change at a time. When the NEMs have similar time scales, the simplest approach is to select a NEM randomly. A more optimized way is to select the NEM that leads to the highest global performance (utility). This means that NEMs must be able to predict the effect of their actions and that the notion of utility must be such that all NEMs do understand how their actions can affect it through their parameter changes. When NEMs have different time scales, one approach is to let the NEMs that optimize rather infrequently (or take longer to converge) to converge first and then allow the other NEMs operate till convergence subject to the constraints set by the first type of NEMs. This approach does require though that NEMs must be able to converge to some fixed parameter values -that the other NEMs ideally should be able to keep intact and still operate adequately towards their own objectives- and in reasonable time, so all NEMs get the chance to operate and perform their optimizations.

Finally, centralized multi-objective optimization deals with NEMs that operate at the same time scale and tries to optimize a global/weighted utility function which combines the utility functions of all considered NEMs. One strong requirement in this case is that all NEMs must be using some explicit utility/objective function for

optimization purposes and that their combination is a feasible task. Contrary to synchronous control theory approaches, the task of global utility function optimization here is delegated to a centralized entity; it can also be one NEM that acts as such entity with the other NEMs delegating to it the task of finding parameter values on their behalf. The latter case is possible if all NEMs manage the same network parameters, as such one NEM alone can set them to values "on behalf" of the other NEMs. In all cases though, contrary to synchronous control theory, there can be only one decision making entity.

Table 6 summarizes the instantiations of Core Mechanisms presented in this deliverable in order of presentation in the rest of this chapter (acronyms will be resolved in the corresponding sections)

Table 6. Core Mechanism instantiations presented in this deliverable.

Instantiation name	Core Mechanism type
ICIC and CCO Coordination	Centralized multi-objective optimization
Load Balancing in HetNets with Relay Stations	Hierarchical optimization
Coverage Self-optimization	Synchronous control theory
Orchestration of Resources and functions in Edge Networks	Centralized multi-objective optimization
Coordinated Link and Node Load Balancing for Virtualized EPC	Centralized multi-objective optimization

3.3 Inter-Cell Interference Coordination (ICIC) and Coverage and Capacity Optimization (CCO) Coordination

The work presented here represents an instantiation of the centralized multi-objective optimization type of Core Mechanisms.

3.3.1 Context of the work

Currently, there has been considerable industrial and research interest to develop competitive solutions in mobile business in order to respond to the expanding demand for cost-effective broadband wireless access while offering enhanced user experience. Self-Organising Network (SON) concepts have been introduced in LTE standardization in order to both provide higher network performance and reduce the operational expenditure (OPEX) for operators. These simultaneously operating and even conflicting SON mechanisms need to be coordinated in order not to cause any instabilities in the network operation. Although this coordination should be done as much as possible in an autonomic way, the operator should have the possibility to intervene in the autonomic loop by setting goals and enforcing them on the network. In this concept, an operator-governed SON coordination is achieved.

Among the proposed 3GPP SON use cases [38], ICIC and CCO are of utmost importance, since, in OFDMA-based networks, inter-cell interference is the main factor hindering the achievement of the high rate requirements, while throughput maximization jointly with QoS provisioning remain among the more significant operator goals. The authors in their previous work in [39] propose a distributed, autonomic, context-aware mathematical framework for ICIC, while investigating in parallel the state of the art. CCO in [40], which is based on central coordination and distributed Gibbs Sampling, optimizes the downlink transmission power of each cell by using user equipment measurements and information exchange among neighbouring cells. The work in [41] proposes a fully distributed and autonomous CCO, without any signalling overhead and human intervention, that uses fuzzy reinforcement learning techniques in order to adjust the down-tilt angle. However, these SON mechanisms are typically studied in isolation without taking into account any possible coexistence.

The SOCRATES FP7 project [42] intended to develop both solutions for single self-organisation (self-configuration, -optimisation, -healing) use cases and an integrated solution, in which the coordination of single SON functions is addressed. In this direction, 24 use cases were identified, from which 10 were selected for further investigation. Although both "interference coordination" and "capacity maximization" as well as "coverage maximization" were included in the first list [43], only "interference coordination" and "coverage hole detection & compensation" were finally selected [44]. Moreover, a SON coordination framework is

introduced [45], but it is demonstrated in an experimental -and not analytical- handover parameter optimization and load balancing coordination. Another work in [46] proposes an experimental system that realizes SON coordination based on flexible policy based decisions; however, it presents exclusively CCO functions. In general, no specific work has been performed for a policy-based ICIC and CCO coordination.

The target of this work is to find the appropriate OFDM resource and power allocation that, when the two SON mechanisms are operating simultaneously, coordinates them in order to guarantee the desired network operation without conflicts or harmful interactions.

3.3.2 Content of the work

In the following analysis, the downlink of an LTE, OFDMA-based system is considered. We focus on downlink in our study due to the related broadband services, which pose higher rate requirements than those in uplink. The topology consists of one "target" cell and *C* neighbouring interfering cells. Let us assume that *N* denotes the number of active users in the target cell and *S* is the number of total available Physical Resource Blocks (PRBs). Moreover, we select $n \in [1,\ldots,N]$ to represent a user, $s \in [1,\ldots,S]$ to represent a PRB and $c \in [1,\ldots,C]$ to represent an interfering cell. The Signal to Interference-plus-Noise Ratio (SINR) of user *n*, who is served by cell *i* over the PRB *s*, can derive from the following equation:

$$SINR_{n,s} = \frac{g_{n,s,i} K_{n,s} P_{s,i}}{\sum_{c=1}^{C} I_{s,c,n} + N_p} = \frac{g_{n,s,i} K_{n,s} P_{s,i}}{\sum_{c=1}^{C} g_{n,s,c} v_{s,c} P_{s,c} + N_p} \quad (4)$$

where $g_{n,s,i}$ is the channel gain between the user *n* and the eNodeB *i* over the PRB *s*, $P_{s,i}$ is the total transmission power at which eNodeB *i* can transmit PRB *s*, $I_{s,c,n}$ is the received interference power at the PRB *s* of the user *n* that is caused by the interfering cell *c* and N_p is the received thermal noise power. Moreover, $v_{s,c}$ is the transmit power coefficient of the PRB *s* from the eNodeB *c*, selected from a discrete set of values [0:*st*:1], where *st* is the step of discrete value selection. Finally, $K_{n,s}$ is the resource and power allocation array of the target cell *i*, which represents the parameters under investigation. It is an *N*S* array that determines which PRB is allocated to which user and the power level of the PRB. $K_{n,s}$ may be expressed as follows:

$$K_{n,s} = \begin{cases} v_{s,i}, \text{ if PRB } s \text{ is assigned to user } n \\ 0, \text{ otherwise} \end{cases} \quad (5)$$

where S_n is the set of PRBs that are assigned to user *n* and $|S_n|$ denotes the cardinality of the set. The provided throughput to the user *n* is equal to:

$$R_n = \sum_{s \in S} \Gamma\left(SINR_{n,s}\right) \quad (6)$$

where Γ is a step function that can be obtained by link level simulations and describes the mapping of channel quality to the expected throughput [47].

In [39], the authors introduced the array $m_{s,c}$ to denote if the subcarrier *s* is used in the interfering cell *c*. In this study, we use PRBs instead of subcarriers for consistency with 3GPP. The array $m_{s,c}$, which may be retrieved by the Relative Narrowband Transmission Power (RNTP) messages, is formed as follows:

$$m_{s,c} = \begin{cases} 1, \text{ if PRB } s \text{ is used by cell } c \\ 0, \text{ otherwise} \end{cases} \quad (7)$$

The total interference caused at the target cell is given by:

$$I_{inter} = \sum_{n=1}^{N} \sum_{s=1}^{S} \sum_{c=1}^{C} I\left(k_{n,s}\right) m_{s,c} I_{s,c,n} \quad (8)$$

where *I()* is a function that converts numeric values to logical, since only the resource (and not the power) allocation of the target cell is required for the computation of I_{inter}.

ICIC and CCO problem formulations

Based on the above analysis, two different SON problems are investigated in this section:

1. *ICIC: Find the appropriate resource allocation in the target cell, in order to minimize the interference caused at the target cell's users.*
2. *CCO: Find the appropriate resource allocation in the target cell, in order to maximize the throughput in the target cell (capacity optimization), while its users experience acceptable channel quality (coverage optimization).*

As denoted before, the parameters to be optimized in both ICIC and CCO are the elements $K_{n,s}$ of the array *K,* which represents the resource and power allocation array of the target cell *i*. Each element $K_{n,s}$ determines which PRB is allocated to which user and the power level of the PRB. However, for ICIC, only the PRB allocation finally counts due to the existence of the logical function in equation (8) that participates in the ICIC objective function (see Table 7 below).

Although both problems share the same parameters under investigation, namely the resource allocation array $K_{n,s}$, they have different objectives (goals). The ICIC and CCO problem formulations are depicted in Table 7. The constraint (11) satisfies that the allocated PRBs in the target cell will not exceed the number *S* of total available PRBs. The constraint (12) is used to represent that each PRB is allocated to only one user of the target cell, while the constraint (13) guarantees that each user will be allocated the appropriate resources, in order to satisfy his rate requirements r_n. The constraint (10) satisfies that each user of the target cell is served with appropriate spectral efficiency SE_n, which is above a certain threshold SE_{thres} [42].

Table 7. ICIC and CCO problem formulations

	ICIC	CCO
Objective Function	$\min I_{inter}$ where I_{inter} is given by (8)	$\max \sum_{n=1}^{N} R_n$ (9) where R_n is given by (6)
Constraints	(11)(12)(13)	$SE_n = \frac{R_n}{S_n} \geq SE_{Thres}, \forall n \in [1, N]$ (10) $\sum_{n=1}^{N} \sum_{s=1}^{S} I(k_{n,s}) \leq S$ (11) $\sum_{n=1}^{N} I(k_{n,s}) \leq 1, \forall s \in [1, S]$ (12) $R_n \geq r_n, \forall n \in [1, N]$ (13)

Note here that the use of the term "objective function" is consistent with the optimization theory, namely the function to be optimized (minimized/maximized). The objective function is related to the definition of the corresponding SON use case e.g. minimization of the Inter-Cell Interference for the Inter-Cell Interference Coordination. Sometimes, these use cases act as intermediate steps to achieve the actual operators goals/ objectives, e.g. by means of controlling the inter-cell interference in a cell, the operator intends to control the throughput of the cell, which stands for the actual gain for both the operator and the end user.

ICIC and CCO coordination problem

The ICIC and CCO coordination problem is investigated as a multi-objective optimization problem with the following vector of objectives to be minimized:

$$\min \left[OF_{ICIC}\left(K_{n,s}\right), -OF_{CCO}\left(K_{n,s}\right) \right] \quad (14)$$

subject to the constraints (10),(11),(12) and (13). The negative sign before (9) in (14) is required to set the whole multi-objective function as a minimization problem. It must be noted that in such problems, including also competing objectives, there is no unique solution but a set of non-inferior solution points, i.e. in which an improvement in one objective requires a degradation of another, which are called *Pareto optima*.

Policy-based ICIC and CCO coordination problem

The previous problem leads to a set of solutions. However, the operator needs to either decide over one solution or give priority to one of the objectives. For this reason, the policy-based ICIC and CCO coordination problem is introduced as a weighted multi-objective optimization problem, as follows:

$$\min\left[w_1 * OF_{ICIC}\left(K_{n,s}\right) + \left(-w_2 * OF_{CCO}\left(K_{n,s}\right)\right)\right] \quad (15)$$

subject to the constraints (10),(11),(12) and (13). The weights $w = [w_1, w_2]$ represent the importance that the operator gives to each objective. They are positive and should sum up to unity, i.e. in this case $w_1+w_2=1$. The weighted multi-objective optimization converges to one optimal solution for a specific selection of weights.

Results

In order to evaluate the operator-governed ICIC and CCO coordination, indicative simulations have been carried out. An NSGA-II (Non-dominated Sorting Genetic Algorithm) algorithm has been used for the solution of the corresponding weighted multi-objective optimization problem. Both the optimization problem (objective function and constraints) and the NSGA-II genetic algorithm (to solve the aforementioned optimization problem and to find the optimal solution) are implemented in Matlab. Here, a snapshot of our simulation is presented, namely the resource allocation in one slot. The simulation parameters are denoted in Table 8. For simplicity reasons, only the first cluster around the target cell is assumed to cause significant interference in the target cell, thus C=6. Initially, we consider 12 users with different QoS requirements (8 with 256 kbps and 4 with 512 kbps, SE_{thres}=0.1 bps/Hz) uniformly distributed in the target cell and st=0.33. These users are scheduled to transmit in the specific slot under investigation.

Table 8. Simulation parameters.

Site-to-Site Distance	500m
Cell Radius	250m
System Bandwidth	5MHz
PRBs	25 with a spacing of 180 KHz
Path-Loss Model	128.1+37.6*log10 (D), D: distance in km
Shadowing	Long-normal with std. dev. 8dB
Thermal Noise Density	-174dBm/Hz
Max Tx Power at eNodeB	43dBm
Tx/Rx Antenna Gain	18dB/0dB

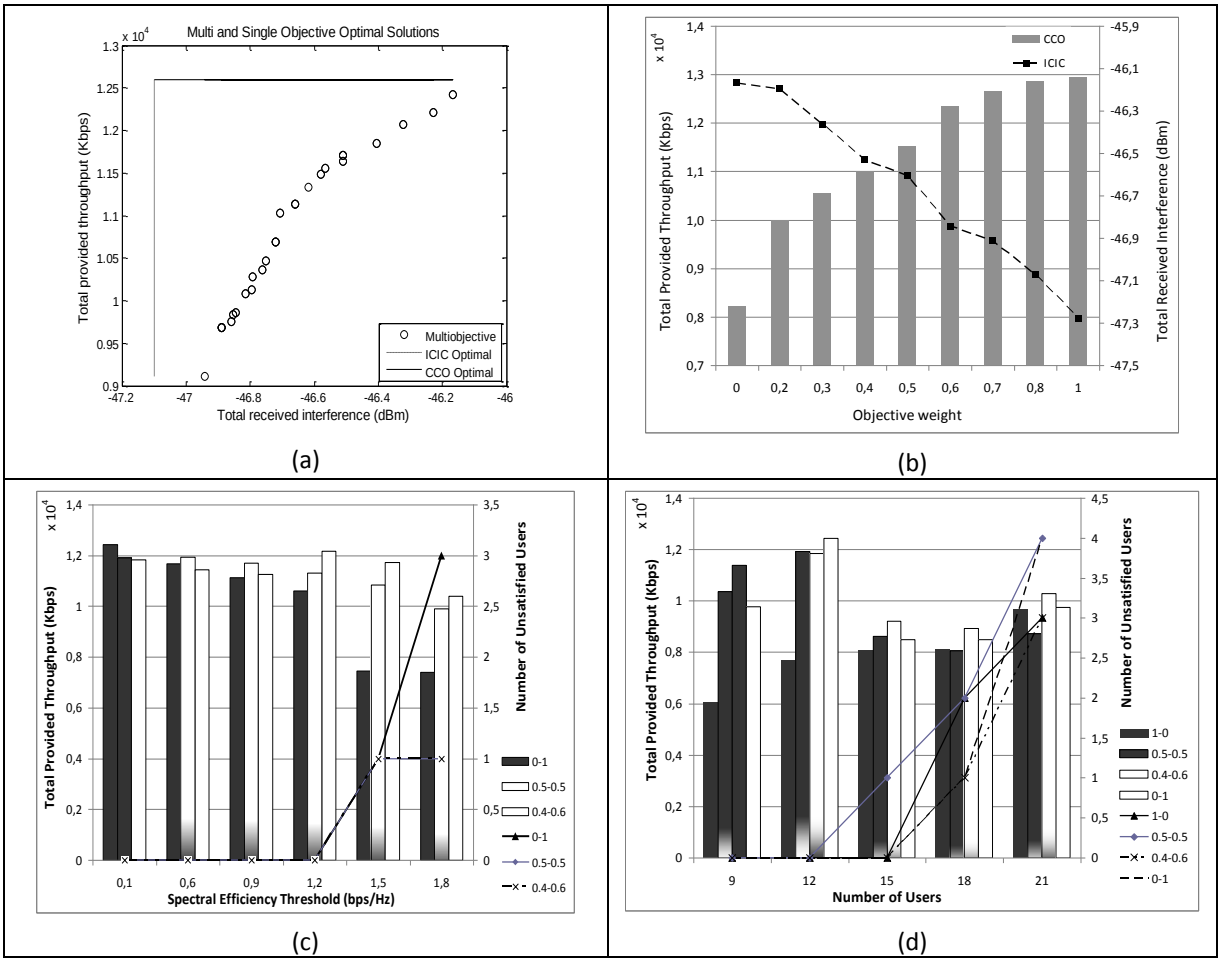


Figure 13. (a) Pareto optima of ICIC and CCO coordination in comparison with the individual ICIC and CCO optimal solutions, (b) Optimal solutions of the weighted multi-objective optimization versus the operator weights, i.e. ICIC versus w_1 and CCO versus w_2, (c) robustness of 3 methods in terms of throughput and unsatisfied users versus SE_{thres}, (d) robustness of 4 methods in terms of throughput and unsatisfied users versus load increase.

The parameters $K_{n,s}$ that determine which PRB is allocated to which user and the power level of the PRB are the outcome of the optimization problem solution. The eNodeB sets the PRB allocation and the PRB powers according to this outcome. The users are served by the target cell's eNodeB (since we investigate the PRB/power allocation in this cell) and the optimal solution satisfies the minimum guaranteed QoS requirements of all the users in terms of rate and spectral efficiency (since these requirements are set as constraints). The coverage optimization is satisfied by the consideration of the appropriate spectral efficiency per user. Each user is allocated with the appropriate PRB/power allocation, so that all their requirements are satisfied independently of whether they are located in the inner part/border area of the cell. The optimization algorithm takes care of allocating less interferenced PRBs in the border area of the cell.

Figure 13 (a) depicts the Pareto optima found by the multi-objective optimization, in comparison with the optimal solutions of the single ICIC and CCO problems. As expected, the Pareto optima satisfy partially both the objectives, in the sense that when a solution gets closer to the optimum of the one objective, it goes away from the optimal of the second. In Figure 13(b), the optimal solutions of the weighted multi-objective optimization are illustrated versus the operator weights w, i.e. ICIC versus w_1 and CCO versus w_2. Objectives (8), (9) are normalized by the individual ICIC and CCO optimal solutions before applying (15), in order to be set in the same scale. It is shown that when the ICIC weight (w_1) decreases, the interference increases, whereas the decrease on w_1 implies an increase in CCO weight (w_2), which in turn gives a higher throughput. In this way, we prove that the operator is provided with the means to trade-off between the different objectives using weights to declare his goals. Figure 13(c) investigates the robustness of three selections (w_1, w_2) in terms of throughput and unsatisfied users when user requirements with respect to SE_{thres} increase. While single CCO gradually

deteriorates, the selection (w_1, w_2) = (0.4, 0.6) gives the more robust solution. Finally, in Figure 13(d), the robustness of four selections (w_1,w_2), in terms of throughput and unsatisfied users when the load with respect to the number of users *N* increases, is presented (starting with 6 users with 256 kbps and 3 with 512 kbps; in each step, 2 users with 256 kbps and 1 with 512 kbps are added). The selection (w_1, w_2) = (0.4, 0.6) also achieves in general the higher throughput and the lower number of unsatisfied users.

3.3.3 Merit of the work

A business, i.e. in our case a telecom operator, may reap significant benefits from the operator-governed SON coordination adoption. The SON introduction allows for OPEX reduction, since the autonomics reduce significantly the operator efforts for configuration, optimization and healing purposes. Moreover, any need for handling all the potential situations (traffic variation, fault occurrence, mobility and radio conditions) via planning, meaning that the worst (most demanding) scenario has to be considered, is minimized, which also leads to significant reduction of unnecessary over-provisioning of resources, which impacts the cost (Capital Expenditures -CAPEX). SON coordination minimizes the human intervention for conflict resolution that requires strong technical expertise and saves time that would be required for manually (re-)configuring the infrastructure; which manual configuration could also lead to errors and inconsistencies apart from delays. Moreover, the SON coordination may lead to important performance gains, since an uncoordinated operation of SON mechanisms would lead to conflicting actions over the network and in this way to unnecessary waste of resources. These gains may regard either the user side (Quality of Experience) or the network side (Quality of Service) and affect a variety of factors (capacity, coverage, stability, etc.) and metrics depending on the selected SON use cases. In addition, the gains improve the competiveness of the operator in the market. Finally, SON coordination may be used to address current deficiencies and in this way to decelerate required infrastructure investments, thus reducing CAPEX.

3.4 SON Coordination Strategies in LTE-advanced Networks

The work presented here represents instantiations of the hierarchical optimization and synchronous control theory types of Core Mechanisms.

3.4.1 Context of the work

The coordination scenarios investigated here have been motivated by recent evolutions on SON, and particularly on self-optimizing network in 3GPP Releases 8 to 11. Different SON functionalities have been studied and standardized as standalone functionalities. In operational conditions, several SONs can be triggered in each base station, and the problem of possible conflicts and more generally, of instabilities becomes crucial. For example, mobility load balancing (MLB) and mobility robustness (MRO) SONs can both modify the same parameter to achieve two different performance objectives [38]. Instabilities however can occur due to the control system itself even if no apparent parameter conflict is present [37].

The notion of stability/instability is used in different contexts and scientific domains as recalled here briefly. In queuing theory, instability is related to the notion of the system load [48]. When the load of an eNodeB approaches one, a significant degradation in different Key Performance Indicators (KPIs) occurs (e.g. number of elastic traffic users in a cell and blocking rate increases, file transfer time increases and the user mean throughput collapses). In this context, SON algorithms can increase the traffic demand that be served and delay the point of KPIs collapse (see for example [49]). The notion of instability in control theory is different. For example, one can look at the parameter behaviour in the parameter plain, and identify unstable behaviour of the control loop. Figure 14 considers the trajectories of two parameters in the parameter plane. Figure 14(i) and (v) are stable, whereas (iv) is considered as unstable in linear control.

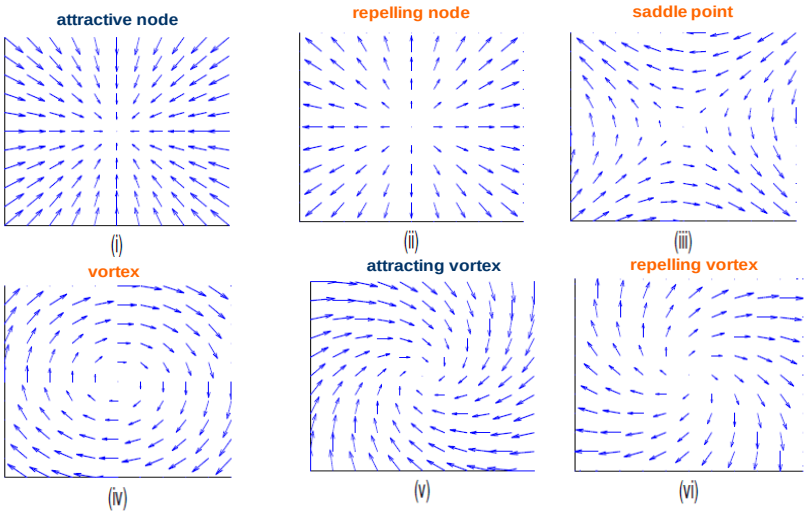


Figure 14. Trajectory of parameters in the parameter plane for two control loops.

This contribution considers the SON coordination problem in two distinct cases:

- The first problem concerns two SON functionalities operating at different time scales, thus creating time hierarchy which provides nice convergence properties. An example can be the dynamic Inter Cell Interference Coordination (ICIC) and mobility Load Balancing (MLB), which adapts the handover margin parameter. The former can vary quickly, in the order of seconds to a minute, whereas cell coverage adaptation that governs the MLB process should vary in a slower time scale to avoid radio link failures related to mobility management.
- The second problem is related to SON coordination having the same (or similar) time scales. A control theory formulation is used to coordinate the different control loops. The work is still on-going, and a simple example demonstrates the approach.

3.4.2 Content of the work

Time hierarchy based coordination of SON

Consider two self-optimizing functionalities (control loops), with different time scales, a fast one (SON_f, f for fast) and a slow one (SON_s, s for slow). The theoretical conditions for the two SONs to converge are:

- The process of SON_f is much slower than the traffic dynamics (frequency of parameter changes f_{fast} and frequency of arrival and departures f_λ verify: $f_{fast} / f_\lambda \to 0$)
- The control process SON_f is much faster than that of SON_s (f_{slow} and f_{fast} verify : $f_{slow} / f_{fast} \to 0$)

HetNet with relays case

Consider an LTE-Advanced heterogeneous network (HetNet) with relay stations. In-band relays are used, namely stations and backhaul share the same frequency resources which are multiplexed in time. Denote by NR the number of relays, and by x_s the portion of time allocated to the backhaul. The index *s*, $s > 0$ refers to the relay stations and $s = 0$ to the eNodeB (macro cell). The direct (eNodeB to mobiles and relays to mobiles) links are active a portion $\left(1 - \sum_{s=1}^{N_R} x_s\right)$ of the time. We consider the downlink with elastic traffic. Users arrive randomly according to a Poisson process of intensity λ, and receive a file size σ bits with a mean size of $\mathrm{E}[\sigma] < \infty$. Mobiles leave the network upon service completion, and mobility is ignored. It is noted that the SON algorithms require only the load information which is insensitive to the file size statistics. The loads can be simulated or obtained using measurements. Denote by A_s the surface covered by station *S,* by $R_s(r)$ the station peak rate and by $R_{rel,s}(r)$ the backhaul link *s* capacity with *r* being the position. We define a quantity related to the loads of the station and backhaul, $\bar{\rho}_s$ and $\bar{\rho}_{rel,s}$, denoted as ρ_S and $\rho_{rel,s}$ respectively (more details available in [36]).

$$\rho_s = \int_{A_s} \frac{\lambda(r)}{R_s(r)} dr \,,\ \rho_{rel,s} = \frac{\int_{A_s} \lambda(r) dr}{R_{rel,s}} \tag{16}$$

From (16) one can calculate the corresponding loads:

$$\bar{\rho}_s = \frac{\mathrm{E}[\sigma]\rho_s}{1-\sum_{s'=1}^{N_R} x_s} \,,\ \bar{\rho}_{rel,s} = \frac{\mathrm{E}[\sigma]\rho_{rel,s}}{x_s} \tag{17}$$

SON models

Relay stations can improve the system performance by serving part of the macro cell traffic, typically at cell edge, which consumes significant eNodeB resources. By absorbing cell edge traffic, the relays can alleviate eNodeB load and reduce the cell congestion. Two SON functionalities are used to optimize the system performance: the first one is a Load Balancing (LB) that adapts the coverage zone of the relay stations by adjusting their pilot powers while keeping traffic channels unchanged. The dynamics of the LB-SON is described by the Ordinary Differential Equation (ODE):

$$\dot{P}_s = P_s\left[\rho_0(\mathrm{P}) - \rho_s(\mathrm{P})\right] \tag{18}$$

where P_s is the pilot power of station *s*, and ρ_s - its load. The discretization of (18) defines the SON algorithm:

$$P_s[n+1] = P_s[n]\left(1+\varepsilon_n\left(\rho_0[n]-\rho_s[n]\right)\right) \tag{19}$$

ε_n being the learning rate, and is chosen here as a constant small number. The properties of the solution, including convergence in the presence of noisy load measurements are obtained using stochastic approximation theorems [49]. In particular it is shown that upon convergence, the solution of (19) balances the relay stations and eNodeB loads, and verifies: $\max_s \rho_s = \min_s \rho_s$. The second SON is a Backhaul Resource Allocation (BRA) functionality. BRA-SON adapts the portion of time x_s allocated to a backhaul link in order to balance the relay load with its backhaul link load. To this end we define *H* as the admissible set for x_s, $s \in \{1,\ldots,N_R\}$:

$$H = \left\{x : x_s \geq 0,\ 0 \leq \sum_{s=1}^{N_R} x_s \leq 1\right\} \tag{20}$$

and denote by $[\cdot]_H^+$ the projection on *H*. The BRA-SON is written as follows:

$$x_s[n+1] = \left[\, x_s[n] + \delta_n g_s\left(\rho[n], x[n]\right)\right]_H^+ \tag{21}$$

$$g_s(\rho, x) = \rho_{rel,s}\left(1-\sum_{s=1}^{N_R} x_s\right) - \rho_s x_s \tag{22}$$

The rationale for (22) is related to the load balancing condition that can be derived from (17):

$$\frac{\rho_{rel,s}}{x_s} = \frac{\rho_s}{1-\sum_{s'=1}^{N_R} x_{s'}} = \frac{\rho_0}{1-\sum_{s'=1}^{N_R} x_{s'}} \tag{23}$$

In the case considered here, the two SONs, LB-SON and BRA-SON, operate at different time scales. The time scale of LB-SON is fixed according to operational constraints (i.e. avoiding too frequent handovers). BRA-SON uses larger time steps to guarantee convergence of the hierarchical system. An equivalent solution for the hierarchical operation consists of choosing the same time periodicity for the two SONs, with constant learning rates: $\varepsilon_n = \varepsilon$ and $\delta_n = \delta$ which are chosen as small numbers verifying $\delta/\varepsilon \to 0$ This solution is equivalent to

choosing $\delta = \varepsilon$ and activating LB-SON and BRA-SON with time periodicity of T_ε and T_δ respectively, satisfying $T_\varepsilon / T_\delta \to 0$. Convergence in distribution to a local optimum solution can be shown. We refer to this solution as synchronous hierarchical mode of operation.

Numerical results

Table 9. System Parameters

Cell layout	One macro cell with four relay stations
Antenna type	Omnidirectional
Cell Radius	2km
Access technology	OFDMA
Fast-fading model	Rayleigh
N_{RB}	10
Resource block size	180kHz
Base station transmit power	46dBm
Relay station maximum transmit power	30dBm
Thermal noise	-174dBm/Hz
Path loss model	$128+37.6\log_{10}(d)$ dB, d in Km
File size	10Mbytes

The network setup comprises one macro cell with a single eNodeB and four relay stations located close to the cell edge. The downlink is considered with elastic traffic. Users arrive randomly according to a Poisson process of intensity λ, and receive a file of size of 10 Mbytes. Mobiles leave the network upon service completion, and mobility is ignored. The coordination algorithm of the two SONs has been tested on an LTE-Advanced dynamic simulator for 30.000 time steps (representing seconds). When the two SON functionalities are activated the LB-SON gradually increases the relays' pilot power. The relays absorb more traffic and see their load increase while alleviating the eNodeB load. At the same time, the BRA-SON adapts the resources allocated to the backhaul links. The joint operation of the two SON functionalities balances the loads of all links (direct and backhaul links). The upper and lower curves in Figure 15 show the worst direct load (namely that of the eNodeB) and the worst backhaul link load (among the four backhaul links), respectively, without SON. It is recalled that the station with the worst load determines the cell capacity. When activating the two SON functionalities, and upon convergence, these two loads are fully balanced (see middle two curves). The maximum link usage is reduced to slightly below 60 percent. As a result, saturation of the macro cell is delayed, giving room to more traffic in the cell. The coordinated SONs successfully balance all the system loads by adapting both backhaul resources and relays' coverage, which on average are increased.

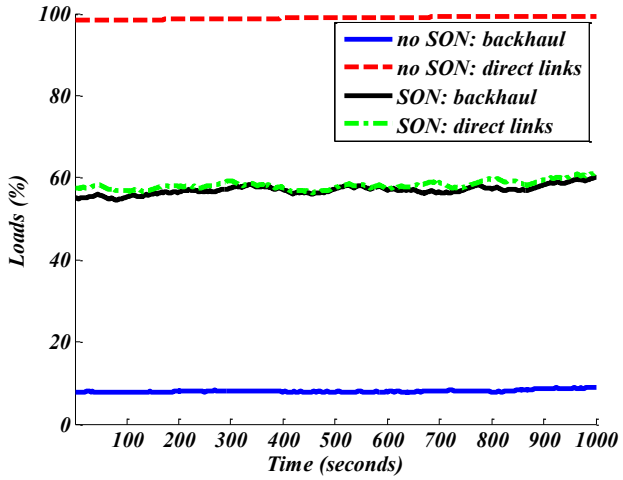


Figure 15. Link usage for the worst direct (station to mobile) and backhaul links with (two middle curves) and without (top and bottom curves) SON.

We define cell edge users as users with the 10 percent lower throughputs in the cell, and compute the average cell edge throughput. Figure 16 presents the cell edge throughput without and with coordinated SONs respectively. Without SONs, average cell edge throughput varies from 0.4 to 0.5 Mbps, whereas when activating the hierarchical operation, 2.1 Mbps average cell edge throughput is achieved.

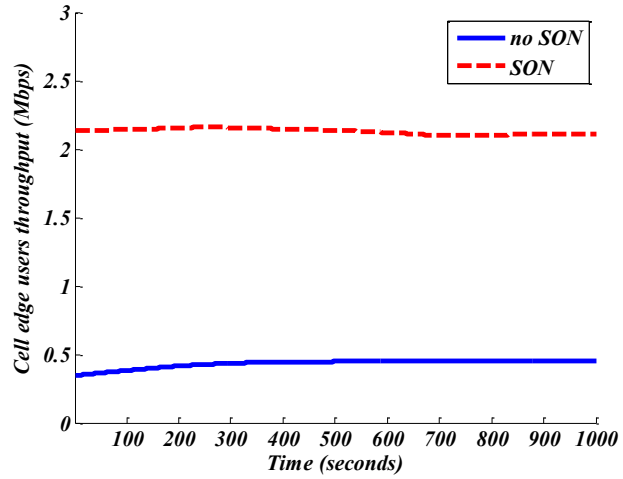


Figure 16. Cell edge thoughput with (dashed line) and without (solid line) SON.

Coordination of SONs operating at the same time scale

We consider here SON functionalities operating at the same time scale. Often, KPIs considered in communication networks (average block call rate (BCR)/throughput/file transfer time) vary smoothly as a function of the parameters and can be locally linearized as a function of the parameters. Denote the vector of parameters controlled by the SON function by θ, and by F a function of the KPIs. A SON function controls one parameter with the aim of optimizing one (possible aggregated) KPI. The control or self-optimizing problem can be written as:

$$\dot{\theta} = F(\theta) \tag{24}$$

If F is linearized, it can be written using its Jacobian $JF(\theta)$, $F(\theta) \approx JF\left(\theta^*\right)\left(\theta - \theta^*\right)$, and in matrix form:

$$\dot{\theta} = A\left(\theta - \theta^*\right) \tag{25}$$

where $a_{ij} = \partial F_j / \partial \theta_i$ represents the derivative of *KPI*$_j$ with respect to the parameter *i* and θ^* can represent a target threshold vector (that in certain formulations can be taken as zero). Solving for θ gives $\theta(t) = \theta^* + e^{At}\left(\theta(0) - \theta^*\right)$, implying that the eigenvalues of *A* must be all negative to ensure stability.

We define the square weighted error:

$$V(\theta) = W\left|A\left(\theta - \theta^*\right)\right|^2 = \left(\theta - \theta^*\right)^T A^T W\, A\left(\theta - \theta^*\right) \tag{26}$$

where $\{w_i\}$ are weights given to the different KPIs associated to the different SON functions and $W = diag(w)$. *V* is a Lyapunov function, representing a global utility of the sub-network that is self-optimized and coordinated. Hence coordination is achieved by the following the gradient of −*V*:

$$\dot{\theta} = -\nabla V(\theta) = -A^T W A\left(\theta - \theta^*\right) \tag{27}$$

We define a matrix *C* as $C = -A^T W$, and the coordination SON mechanism:

$$\dot{\theta} = CA\left(\theta - \theta^*\right) \tag{28}$$

The elements of the matrix a_{ij} are given by $a_{ij} = \partial F_j / \partial \theta_i$ namely the derivative of a (function of) KPI *j* with respect to parameter θ_i controlled by SON *i*. It is noted that in this formulation, two instantiations of the same SON functionality (i.e. one distributed/composite NEM instantiation in the UMF vocabulary) are considered as distinct SONs. However the approach remains valid if applied on different NEM instantiations (multiple instantiations of different SON functionalities). Equation (27) is of gradient type for all the coordinated functionalities (NEMs). The calculation of the derivatives of $\partial F_j / \partial \theta_i$ can be done efficiently using techniques such as simultaneous perturbation analysis using "one shot" calculation per matrix. We note that the formulation above is fully distributed. It is noted that other formulations of the problem can be considered which do not involve the construction of the Lyapunov function V. An example is the Convcave N-Person games of Rosen [50].

Coverage self-optimization case

The case described hereafter showcases the control approach for coordination considering one SON functionality instantiated in many eNodeBs. The application of the approach to different SON functionalities per eNodeB is still under development within the project. We consider a SON functionality which self-optimizes coverage. The SON functionality adapts traffic channels' powers at a time resolution of a few seconds, namely the power of the entire band is adapted (not per sub-channel). Let A_i denote cell *i* and $|A_i|$ its area. $R_i(r)$ is the data rate at location *r*, and the mobile is considered covered if its data rate exceeds a minimal bit rate R_{min}. The coverage probability of cell *i* is given by:

$$K_i = \frac{1}{|A_i|}\int_{A_i} 1_{\{R_i(r) \geq R_{min}(r)\}} dr \tag{29}$$

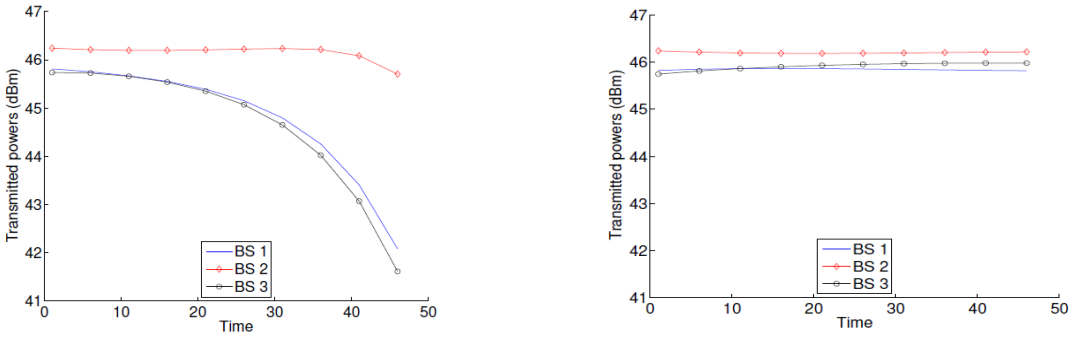
Define the set of neighbours of eNodeB *i* by B_i, and the coverage probability a neighbour *j* of *i* by G_i :

$$G_i = \frac{\sum_{j \in B_i} |A_j| K_j}{\sum_{j \in B_i} |A_j|} \tag{30}$$

The traffic model is the same as in the previous case (downlink is considered with elastic traffic with users arriving randomly according to a Poisson process of intensity λ, and receiving a file of size of 10 Mbytes). A hexagonal network with 12 eNodeBs is considered using a wrap-around model. The matrix *C* in (28) is defined as $C = -\left(JG\left(P^*\right)\right)^T$, namely the Jacobian of G at P^*, with the vector P^* being the vector of the eNodeBs' powers.

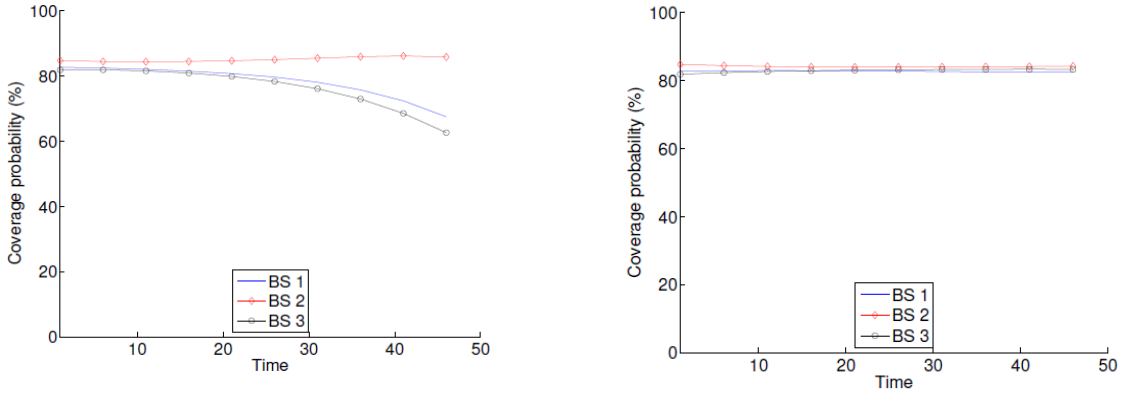
Results

We choose a probability target of $\overline{G_i} = 80\%$ for all eNodeBs. The evolution of the traffic channel powers is shown in Figure 17 for three eNodeBs, without coordination (case (a)), and with coordination (case (b)). Similarly, in Figure 18 the probability of coverage is shown in the non-coordinated case (a) and the coordinated case (b). One can see that when the coordination mechanism is not used, the system becomes unstable and as a result, coverage performance collapses. Hence the coordination mechanism stabilizes the coordination process.



(a) No coordination (b) With coordination

Figure 17. Power evolution of the non-coordinated (a) and of the coordinated case (b).



(a) No coordination (b) With coordination

Figure 18. Coverage probability for the non-coordinated (a) and for the coordinated case (b).

3.4.3 Merit of the work

This work proposes methods for coordination of self-optimizing functionalities operating at different and at the same time scales. The merit of SONs in LTE networks has been extensively shown in many studies from both industry and academia. Coordination mechanisms are necessary to achieve the gain from SON functionalities and guarantee the stability of the network. This work has provided candidate solutions for SON coordination for two cases of particular interest: in the first, the SON functionalities operate at different time scales and in the second, the SON functionalities operate at the same time scale.

3.5 Orchestration of Resources and Functions in Edge Networks

The work presented here represents an instantiation of the centralized multi-objective optimization type of Core Mechanisms.

3.5.1 Context of the work

This work targets the edge of traditional networks. Network evolution is moving towards the deployment of heterogeneous Edge Networks where resources and functions are dynamically instantiated, moved and orchestrated by the Network Operator to serve Users' services demands.

In fact, it is argued that paradigms as SDNs (Software Defined Networks)[1] and NFV (Network Functions Virtualization)[2] could express most of their "disruption" at the edge, pushing more and more the migration of processing power and storage capabilities in that direction. This migration will create a growing "complexity" at the edge (number and heterogeneity of nodes) which will require introduction of global orchestration capabilities integrated with local autonomic-cognitive functions.

In a reference scenario, as shown in Figure 19, each Edge Network connects to an Edge Node, which is a node interconnecting the Edge Network with the Core Network (e.g. an Edge Node can be seen as a future evolution of a Broadband Network Gateway (BGN) node which is including also powerful processing and storage features).

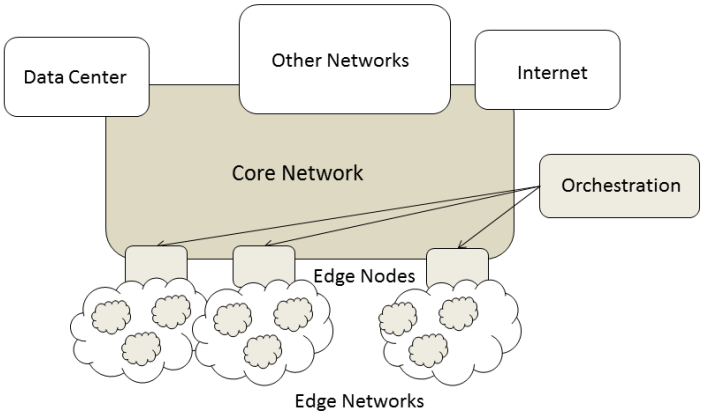


Figure 19. Schematic description of an "edge networks" setup.

Edge Nodes have virtual resources for running network functions (fully developed in software) and services demanded by the corresponding Edge Network. Examples of network functions are: Deep Packet Inspection (DPI), Broadband Network Gateways (BNGs), Network Address Translation (NAT), firewalls, application acceleration over WANs, Session Border Controllers (SBCs), IPsec or SSL VPN gateways, SLA monitoring, Network monitoring, QoS measurement, Intrusion Detection Systems (IDSs), Load Balancers, etc.

Depending on the service demands dynamics, virtual network functions can be properly instantiated, moved and orchestrated across all the Edge Nodes, which are creating a sort of pool of available processing, storage and networking resources. The orchestration block, on the right side of Figure 19, is in charge of orchestrating the Edge Nodes resources and functions, locally controlled by sets of control loops (or NEMs).

[1] *"In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from applications" (Source ONF – White Paper).*

[2] *"Network processing functions can be developed in software, running on standard hardware and that can be instantiated and moved in various locations in the network" (Source NFV – White Paper)*

3.5.2 Content of the work

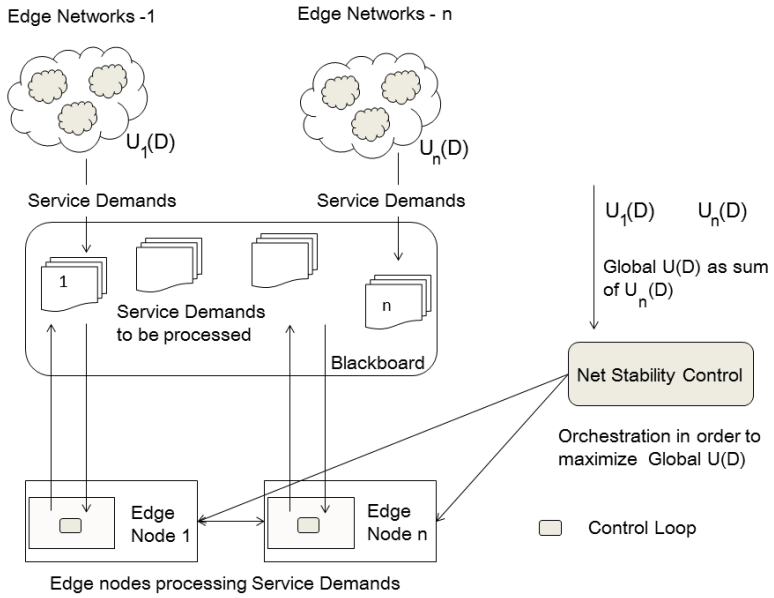


Figure 20. Schematic description of the orchestration framework.

The content of the work refers to designing and simulating the orchestrator which is called NEM Network Stability and Control (NSC). In particular the orchestrator has the scope of controlling the functioning of a set of control loops (or generally speaking NEMs) which are running on the Edge Nodes. These control loops -in this specific context- are handling and executing specific network functions or services demanded by the Edge Networks (so to ensure best use of resources and acceptable performance levels).

Figure 20 represents an abstract model. The Users of the Edge Networks are generating Service Demands which have to be processed by the corresponding Edge Node. Each Edge Node has allocated a certain number of virtual resources to serve the Service Demands of its Edge Network; potentially unused resources can be allocated to serve Service Demands of other Edge Networks in case of congestion or hot-spot of the other Edge Nodes, but this requires an orchestration to avoid instabilities. Each Edge Network has allocated a Utility function, which is computed at every sampling time to check the level of the performance with which the Service Demands are covered (Utility functions have very attractive theoretical properties in practical autonomic computing systems [51] [52]). In real instantiations the metric encompasses QoS/QoE parameters associated to the performance metrics (e.g. response time, throughput, etc.) of the Edge Networks (or sub-networks) serving groups of Users according to specific SLAs. A Global Utility function is, in turn, a function of the Utility functions of the Edge Networks (e.g. weighted sum or product).

During each sampling time, the NSC receives, as inputs, the Utility functions of the Edge Networks: NSC has the objective of maximizing a global Utility function. Based on its global knowledge, the NSC computes configurations to be enforced in the control loops (or NEMs) of Edge Nodes in order to maximize the Global Utility function. The parameters which are manipulated are the configuration parameters of the local control loops (or NEMs). In principle this approach brings to an NP-hard discrete configuration problem, and can be solved by a wide variety of standard metaheuristics.

The simulations are dealing with a case where there are three Edge Nodes, which are hubs of three corresponding Edge Networks. Control loops (or NEMs) are instantiated in said three Edge Nodes have to be orchestrated when the values of the Utility functions go below a certain threshold.

In the simulations carried out, the Edge Nodes execution environment is based on the Linda model [53]. Each service demand is labelled as a Linda tuple written on the Linda tuple space, named blackboard, whilst Linda take operation is used by Edge Nodes to offer their execution capabilities. There are three types of service demands (gold, silver and bronze). Linda model provides natively decentralized load balancing capability in

handing the three categories of service demands, i.e. each Edge Node locally decides if it is able to execute the service demands and orchestrates its local resources

NSC orchestration algorithms can be based on different metaheuristics[3]. Simulated Annealing (SA) is one of the explored metaheuristics. The name of simulated annealing originates from the simulation of annealing process of heated solids. "*In condensed matter physics, annealing denotes a physical process in which a solid in a heat bath is heated up by increasing the temperature of the heat bath to a maximum value at which all particles of the solid randomly arrange themselves in the liquid phase, followed by cooling through slowly lowering the temperature of the heat bath. In this way, all particles arrange themselves in the low energy ground state of a corresponding lattice.*" [54] [55].

There is a huge prior-art on SA convergence time, where several customizations are described to improve it, basically depending of the shape of the utility function and Figure 21 shows the flow chart of a generic SA. The algorithm adopted for the simulations is a SA adapted and simplified for optimizing the convergence time, which is a key success parameter for NSC induced orchestration. Other metaheuristics are also under analysis and comparison.

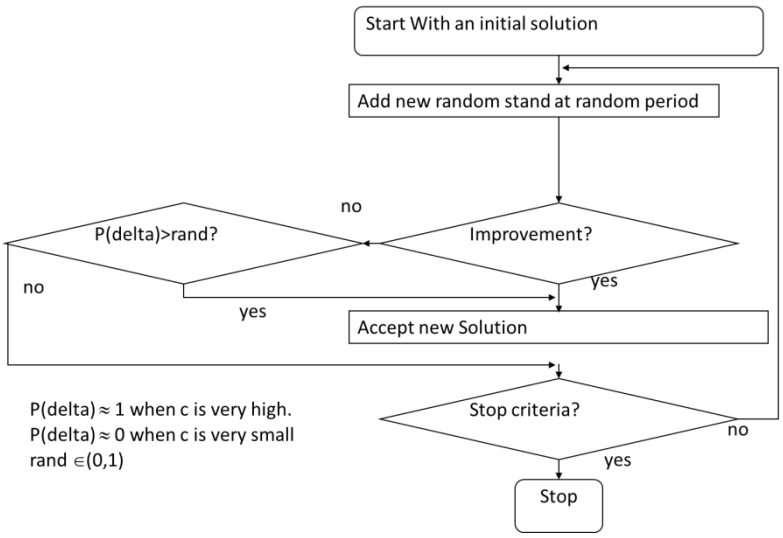


Figure 21. Principles of functioning of the simulated annealing.

Figure 22 is showing a critical case where a number of service demands (service demands are kept intentionally at an abstract level rather than being tied to a specific type of service) generated by three Edge Networks are suffering a delay in the execution (due to poor local resources, for example) and as such there is a degeneration of the QoS performance. In those cases the Utility functions associated to the respective Edge Networks (and the global one) are decreasing. In these circumstances NSC is entering in action calculating a reconfiguration of the control loops parameters in order to re-establish acceptable conditions.

[3] *Solution methods that utilize interaction between local improvement procedures (local search) and higher level strategies to escape local optima and ensure robust search in a solution space*

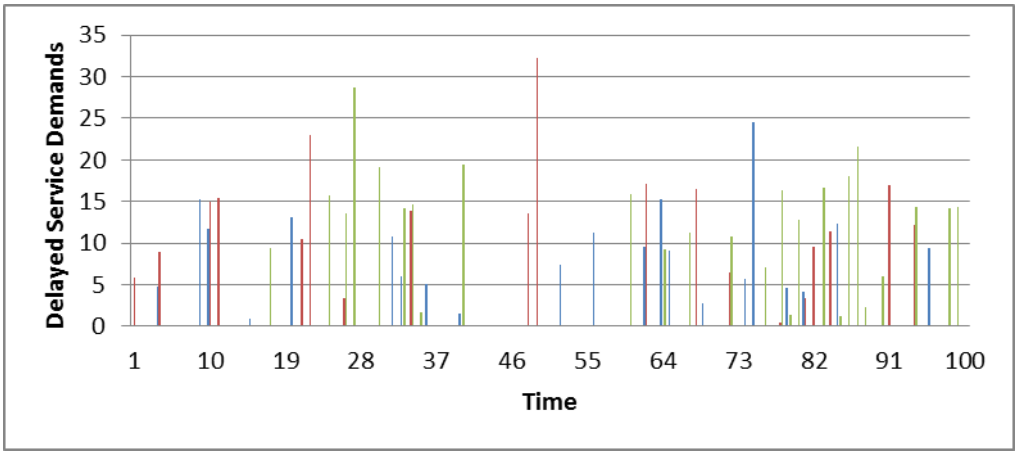


Figure 22. Service demands generated by three edge networks: histogram shows the numbers of delayed service demands with degeneration of QoS performance.

Figure 23 is showing the Global Utility function during a certain time window. Black line shows the advantage of having an orchestration mechanism like the one performed by the NSC.

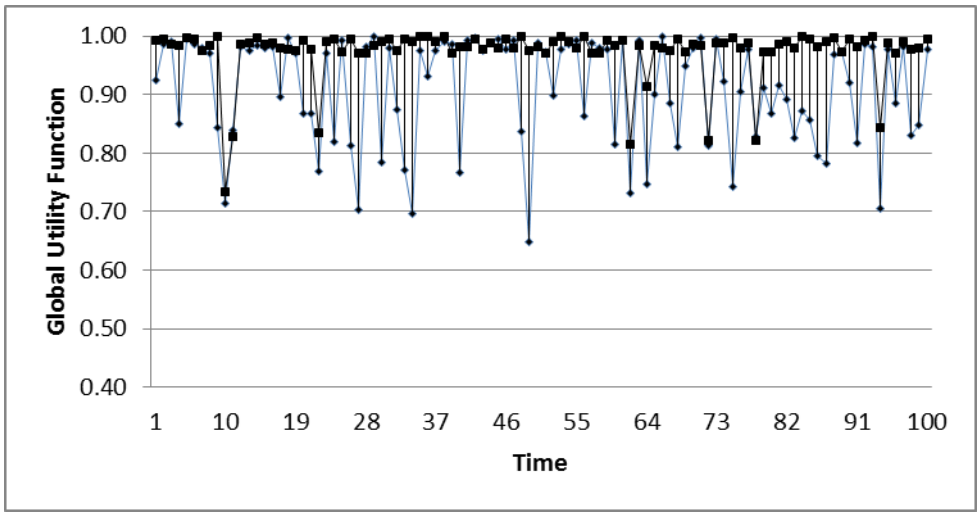


Figure 23. Global utility function: black line with squares in case of orchestration (gray line in case of no orchestration).

3.5.3 Merit of the work

Simulations showed that NSC mitigates the potential network instabilities (reflected by degeneration in the QoS). The approach adopted creates a stable collaborative environment across several Edge Nodes in which dynamic self-adaptation is achieved globally.

The *Poll When Idle* approach provides a native load balancing of local resources (within each Edge Node) in handling three classes of service demands. Moreover the collaborative interactions among Edge Nodes (orchestrated by the NSC) ensure the maximum resource utilization allowing the environment to scale as much as possible using the available resources, not only within but also across all Edge Nodes.

Overall the environment is providing a dynamic self-adaptation behaviour guaranteeing the target performance objectives. NSC uses straightforward orchestration heuristics (SA) and it is easily understandable from a design point of view. Time scales requirements of the NSC orchestration are dependent on the QoS requirements of the services demands and can be controlled by the operator. Convergence time of the heuristic engine should be a point of attention in the design of the NSC; this is an issue for further study.

3.6 Coordinated Link and Node Load Balancing for Virtualized Evolved Packet Core

The work presented here represents an instantiation of the centralized multi-objective optimization type of Core Mechanisms.

3.6.1 Context of the work

Due to the fast pace of technological evolution in the field of wireless access technologies and bandwidth-intensive user applications, the existing telecom networks are facing increasing pressure to meet the QoS/QoE demands of mobile users. At present, most of the content/application service hosting and management is being concentrated at the core. As a result, all the user traffic has to go through the core over the backbone/backhaul and hence a lot of resources (bandwidth and processing wise) are consumed. As is evident, the operators maintain a central data center consisting of extensive storage and processing entities (e.g. Network Management Systems (NMS), Thin Client Servers, Content, etc.). Also, the mobile core network itself (i.e. the Evolved Packet Core (EPC) in case of LTE) is centralized and far away from the access network. Current technical specifications aim at decentralizing the core by assigning local breakout gateways which allow for early breakout of the traffic and are located closer to the access network. Although this is expected to move some traffic from the core network, it is still a static approach in the sense that the local gateways are statically assigned and distributed.

In a previous deliverable (D3.5 [3]), we have introduced a dynamic EPC instantiation approach. The goal is to run the mobile core network functions on top of a virtualization layer on general-purpose hardware called General Purpose Node (GPN), such as smaller and more regionalized data centers, rather than running them on dedicated mission-specific hardware (see Figure 24). Instead of core network functions, services or content being bound to a fixed physical location (as given by the corresponding hardware), they can be flexibly and dynamically instantiated, scaled and migrated at runtime to the most appropriate place in response to varying load conditions and user requirements, such as to reduce the overall load on the operator network. Note that load variations can occur on regional (day/night, commercial/residential) and even national (different time zones) levels. Potential load optimization gains thus clearly depend on the deployment strategy for data centers, the intelligence of the algorithms, transport network topologies, etc. Figure 24 presents a high-level sketch of this concept.

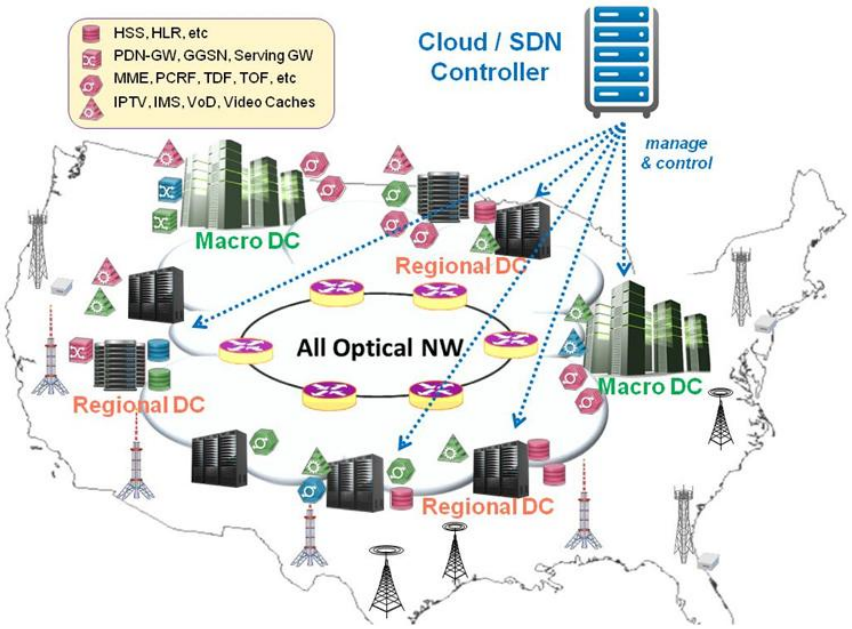


Figure 24. Soft EPC Vision - distributed, dynamic EPC instantiation of GPNs/Data Centers.

3.6.2 Content of the work

Obviously, reducing overall load by intelligent EPC instantiation pertains to both node and link loads. Typically, link load balancing is always expressed in terms of minimizing the maximum link utilization. The rationale for this metric is that, if maximum space is left in each link after all resources have been allocated, there is maximum flexibility for new traffic demands to be routed "capacity-optimally" through the network. If, on the other hand, some links were completely saturated while some links were almost unused, new demands might need to take long detours to circumvent congested areas in the network, leading to suboptimal results in terms of overall network load.

The same rationale can now be used to argue for balanced node loads. Similar to links, node resources (i.e. storage and processing capacity) should also be used across the network in a balanced way. As depicted in Figure 24, a centralized controller (probably part of the operator's Mobile Network NMS) will usually be employed to manage the distribution of EPC functions across the available data centers. As, from the point of view of the mobile network, the transport network is completely transparent, the centralized controller will basically make decisions about function distribution independently of transport network considerations.

However, it is obvious that only a coordinated decision making that considers both node and link loads can make globally optimized decisions where both node and link loads are balanced. There are two options to achieve this. First, a control loop that balances node loads and another one that balances link loads could run independently and be orchestrated via an orchestration engine. This has the advantage that both control loops do not need knowledge about the respective other network domain, which is well in line with the 3GPP specifications where mobile network and transport network are considered completely independent. As a second option, both control loops could be integrated into one common joint optimization control loop with a multi-objective optimization target. In our study, we make a first step towards evaluating the performance benefit of the second option over the first one. To this end, we designed an optimization algorithm that aims at balancing both node and link loads at the same time (i.e. an embodiment of the second option). We compare that with an approach where EPC functions are fixed in a first step, i.e. a priori, and only then paths between base stations and gateways via the respective EPC functions are computed. This intuitively corresponds to the case where node and link load balancing are done independently, i.e. the first option.

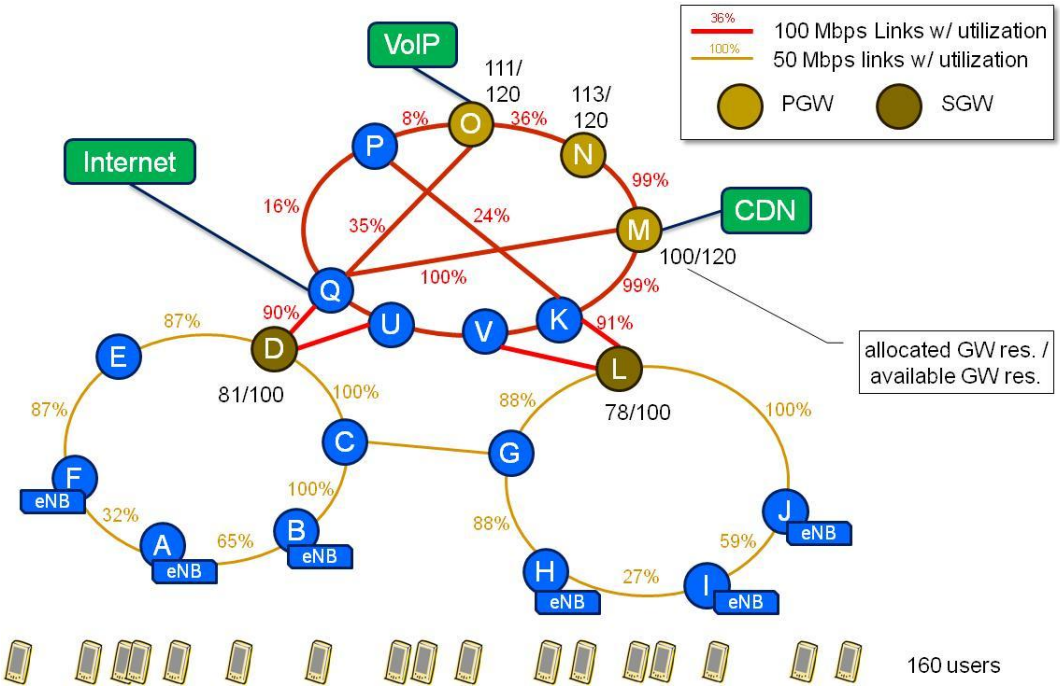


Figure 25. Node and link loads determined independently (Test Case 1).

Figure 25 shows resulting loads for the case that EPC functions are determined first and paths are computed after that (called "test case 1" or "TC-1"). In direct comparison, Figure 26 shows the same topology, but this

time with flexible GPNs (i.e. small regional data centers) instead of the fixed dedicated EPC nodes from the previous example. In this case, the algorithm decides both function placement and paths routing (called "test case 2" or "TC-2").

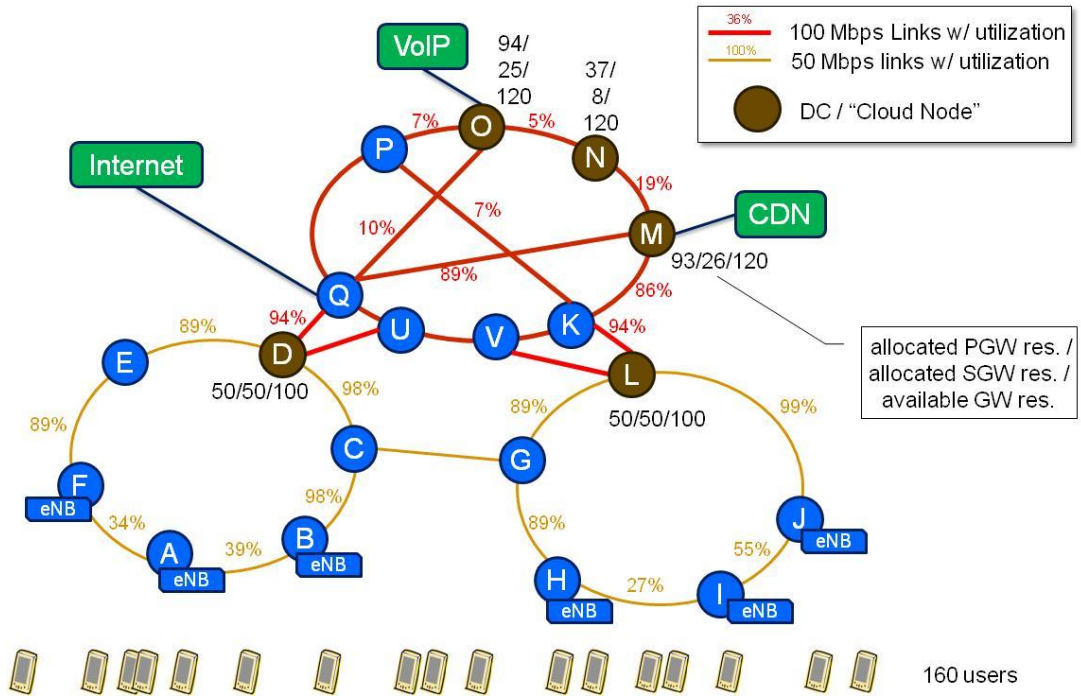


Figure 26. Function placement and routing are determined together (Test Case 2).

Clearly, Figure 25 and Figure 26 are only sample topologies. Yet they are the ones used in our simulations and do represent realistic topologies: more meshed structures in the "core" network and aggregation rings in the aggregation network; eNodeBs would be connected to the aggregation rings via star, tree or chain topologies, but for the purpose of evaluation these can be collapsed into single "ingress points". We assumed a set of 160 mobile users that are uniformly distributed across the eNodeBs. Each user can have between 1 and 3 active services. We distinguish between three service types, namely voice, Internet and video. The latter is assumed to be delivered via an operator-controlled mobile content delivery network (CDN).

While it is evident that the distribution of SGW and PGW functions across GPNs and thus the resulting link loads are very different in both test cases, it is difficult to directly see why either of the two should be beneficial over the other one. To this end, we have depicted the overall bandwidth consumption for both test cases (and two more) in Figure 27. Consumed bandwidth is about 20% less for TC-2, i.e. with joint optimization.

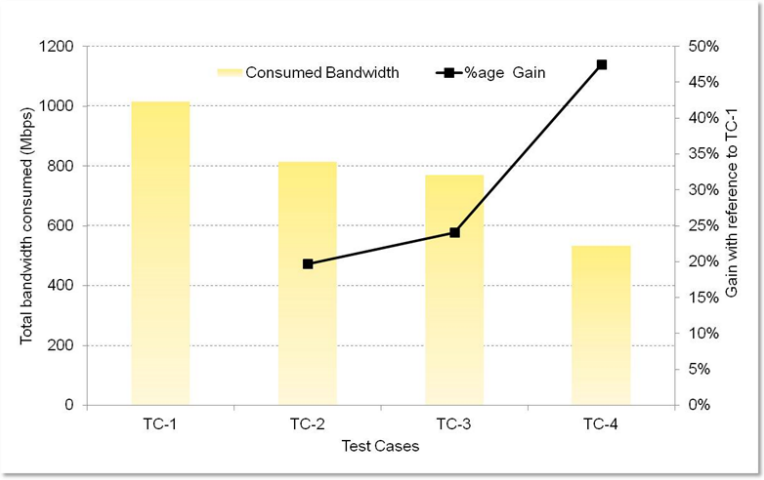


Figure 27. Total consumed bandwidth for various scenarios.

Test Cases TC-3 and TC-4 differ from TC-2 in that even eNodeBs are considered GPNs, i.e. EPC functions can even be placed directly on eNodeB nodes. In TC-4, we also move the CDN gateway from node M to node L and add another CDN gateway to node E. This allows much earlier breakout of traffic, provided that EPC gateway functions are placed intelligently so routing inefficiencies can be avoided. TC-4 is therefore another good example for the kind of performance gain that can be expected with joint optimization of function placement and path routing, i.e. node and link load optimization.

Our placement algorithm is described in [56]. It basically is a greedy algorithm that admits all sessions one-by-one trying to find shortest paths between base stations and gateways with sufficient resources on intermediate links and GPNs. The order in which sessions are admitted is critical, as it is possible to block important GPNs too early, forcing long detours on later admitted sessions. In terms of dynamics, we currently assume changes in the domain of minutes to hours, which corresponds well to regional/local load fluctuations. However, in principle, the algorithm is independent of the desired time scale, which really depends on the geographic scale, for which it is to be applied.

This algorithm, while placing functions in a way that minimizes link loads in the network, is not in itself optimized with respect to an even distribution of EPC functions across GPNs. In other words, while there is joint optimization considering node and link resources, the objective of minimizing the maximum *node* utilization was not considered (we refer to this as unbalanced joint optimization). Thus, we extended that algorithm to improve node load balancing in addition to joint optimization. Figure 28 shows node utilizations for all GPNs (from the scenario in Figure 26) for the original algorithm [56] and the modified / balanced variant. Clearly, the balanced version -while minimizing overall link load equally well- balances node load much better compared to [56] (with a standard deviation of 0.13 vs. 0.36 for node utilization across all GPNs).

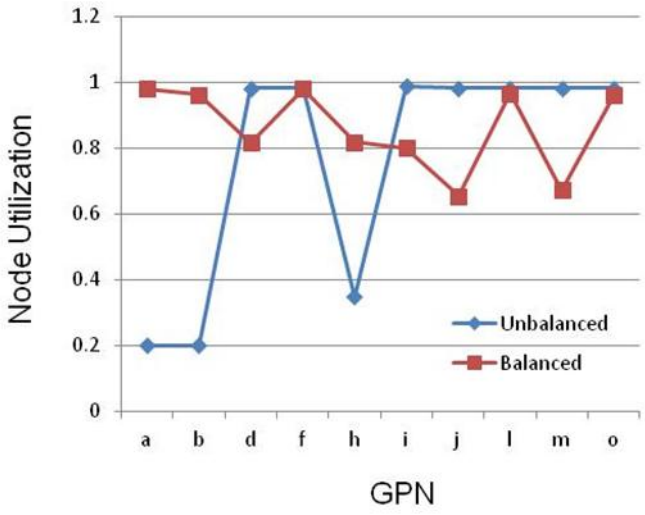


Figure 28. Node utilization for unbalanced and balanced joint optimization algorithms.

3.6.3 Merit of the work

The static and inflexible nature of current mobile networks is apparent and running them on general purpose nodes with flexible and load-aware instantiation of EPC functions across the network is an attractive solution. However, algorithms are required to manage resources in such mobile networks clouds, both in terms of link as well as node loads. Traditionally, these two things have always been considered independently. Particularly in 3GPP mobile networks, the transport network has always been neglected as sufficient transport capacity was simply “assumed”.

We have shown that jointly optimizing function placement under the aspects of node and link load can lead to much better results compared to solutions with function placement a priori, rendering the latter fixed for the sake of link load optimization. We have seen that not only significant link load savings can be achieved, but resources can be allocated across GPNs in a reasonably balanced way at the same time. This makes both nodes and links ready for sudden and unpredictable load increases anywhere in the mobile network.

4 Interactions: Factors and Considerations

For the needs of this section we will focus on the integration aspects involving the COORD core block since it is the one dealing with conflict identification and resolution, and in general with the grouping and alignment in time of NEMs.

Therefore we assume that the starting point is the NEM registration event to COORD [2] which follows the instantiations of NEM through GOV with the NEMs themselves managing the establishment of the proper channels with KNOW for the retrieval of any information they may be needing during their runtime. It is the role of COORD at this stage to try and diagnose potential conflict situations and potentials for joint optimization options based on the available NEM information and assign NEMs to instantiations of COORD Core Mechanisms.

This assignment is in practice implemented by setting constraints to the NEMs' behaviour in terms of when and for how long they can run their optimization cycles, in terms of whether they can consider their original optimization targets or some modified targets in order to take into account the existence of other NEMs, and in terms of modifications they are allowed to perform into the network parameters they control.

4.1 What constitutes a conflict

While defining what constitutes a conflict and addressing it is an on-going activity within the UniverSelf project, there is consensus that the definitions derived in the FP7 Socrates project [42] [57] form a solid starting point. Using similar vocabulary with [57], the two "main" types of conflict are:

- parameter value conflict, and
- metric value conflict

Parameter value conflict is the easier to detect; it refers to the situation where multiple NEMs have control over the same exactly network parameter (e.g. antenna pilot power, link weight setting in OSPF, etc.) As a result there exists the possibility that if such NEMs are left unattended, it may simply lead to the situation where each one of them will be setting a specific value to a network parameter, only to have other NEMs override this setting with their own preferences. In cases this happens it will mean that none of the NEMs will be performing as intended and that there is a high chance of oscillatory behaviours with the network parameter of interest oscillating along a wide range of values.

Metric value conflict refers to the situation where a metric used as input from one NEM is affected by modifications by other NEMs. This can constitute a problem for example if there is a NEM A that uses metric Z (e.g. utilization) as an input and it is using a reinforcement-like type of learning and optimization. This means that NEM A inherently makes the assumption that the actions itself takes are responsible for the evolution of utilization and relies on the observations of utilization in order to learn and adjust its behaviour. If however there is a NEM B that takes actions that affect the utilization, this will mean that the assumptions made by NEM A were wrong. As a result, NEM A will learn and possibly converge to a wrong behaviour; even worse, if NEM B starts behaving differently or is removed by the system altogether, then NEM A will have to again (from scratch in case of reinforcement learning) start learning a new behaviour.

Similar problems can happen in general whenever the changes made by NEM B try to "drift" NEM A away from its nominal standalone behaviour and/or invalidate any assumption that NEM A makes about its inputs. In addition a potential problematic situation may arise when NEM A can cope with any changes to its inputs by NEM B, but the output of NEM A "feeds" NEM B forming a loop; either explicitly/directly or implicitly through a cascade of other NEMs leading eventually back to NEM B.

However metric value conflicts are not straightforward to detect for two reasons. First, there might be a cascaded set of actions by a set of NEMs which actually leads to an input metric of another NEM being affected, as mentioned above. For example it may be a NEM C which affects NEM B which, in turn, affects NEM A, which may also feed back to NEM C. This means that it would not be enough to consider NEM A and NEM B alone as conflicting, but NEM C should be also taken into account. On the other hand though such an approach would lead eventually to considering that "everyone somehow affects everyone else" so there needs to be a "stopping point" in this process. The second reason is that there are exist cases where an input of NEM A being affected by the actions of NEM B do not necessarily constitute a conflict but a normal situation where NEM A is

expected to react to changes in that input. For example, a traffic engineering NEM expects that it will have either periodically or based on some other triggering event (link overload) to redistribute traffic along the available paths of a core network domain. Changes to the link load by another NEM (e.g. a NEM running at the access network and affecting the way the traffic enters/leaves the core domain) are not to be treated as a conflict but rather as a normal situation.

These issues are currently under investigation in WP2 for the needs of the deliverable D2.4 - Unified Management Framework (UMF) Specifications Release 3; here we will present a list of factors regarding the behaviour and specification of a NEM that may be useful when defining whether a situation constitutes a conflict together with the implications they have for performing conflict resolution for the NEMs and Core Mechanisms instantiations presented in this deliverable; that is if they were to be deployed concurrently for the needs of a specific network and service configuration.

4.2 Behaviour and Specification Factors for Conflict Identification and Resolution

In order to identify potential conflicts, according to the definitions provided above and also to consider resolution mechanisms, the following factors in the behaviour and specification of a NEM (and also for a Core Mechanism instantiation since it is really a grouping of NEMs) could be considered.

- **Topological scope**: the domain it addresses is the first obvious factor since it determines the neighbourhood of a NEM/Core Mechanism instantiation, potential overlaps with other NEMs and adjacencies with other domains and NEMs running on them
- **Inputs**: the inputs required by a NEM during its operation so as to be able to identify potential metric value conflicts
- **Parameters manipulated**: the network and service parameters a NEM affects so as to be able to identify parameter value conflicts
- **Metric affected:** the metrics affected by the operation of a NEM so as to be able to identify potential metric value conflicts
- **Dynamicity:** this refers to the intervals between activations of the decision making process of a NEM; useful not only for judging whether a situation actually constitutes a conflict but also for checking the feasibility of a Core Mechanism for conflict resolution
- **Convergence:** this refers first of all to the type of convergence; does a NEM converge to some "fixed" network parameter values or does it converge to a fixed behavioural policy? The latter means that the NEM never really fixes a parameter value but it only fixes the way it reacts based on its perception of the environment. In addition it is useful to know whether a NEM needs a number of iterations every time its decision making process is invoked and whether at each iteration it enforces something to the network. For example "NEM A triggers its decision making process every 30mins and once it does it needs 50 iterations at 1sec intervals; at each iteration there is an enforcement of actions on the network". This is useful for example so as to see whether a separation in time strategy can be applied, by having NEMs converge to some fixed network parameter values.
- **Correlation between iterations/decision making processes:** this refers to whether there exists some kind of feedback loop by means of a reward/utility value between successive iterations; it can define how sensitive a NEM is in "interference" while converging which may help identifying a conflict as well as the feasibility of a "partially overlapping" separation in time strategy
- **Explicit utility function:** this refers to the existence or not of an explicit objective function a NEM is trying to optimize; it can determine whether strategies based on multi-objective optimization can be applied (as well as synchronous/asynchronous control theory approaches)

In the following sections we will present how these factors translate for the NEMs and instantiations of the Core Mechanisms presented in Chapters 2 and 3 respectively followed by a brief discussion on the implications they may have on defining potential conflicts and subsequently assigning conflict resolution mechanisms in the forms of the available Core Mechanisms.

4.3 NEMs and Core Mechanism Instantiations: Relationship with Factors

4.3.1 Hybrid P2P selection for optimized user and network performance

- **Topological scope**: core network segments
- **Inputs**: utilization/load of inter-domain links, domain business relationships, location of available peers
- **Parameters manipulated**: the NEM does not explicitly alter a network parameter, but it generates a list of peers from which a specific content object is to be retrieved
- **Metric affected:** utilization/load of inter-domain links and consequently the traffic load at the domain's ingress routers and intra-domain links
- **Dynamicity:** on a per content object request
- **Convergence:** for each request there is one step
- **Correlation between iterations/decision making processes:** no
- **Explicit utility function:** no

4.3.2 Distributed decision engine

- **Topological scope**: wireless access domains
- **Inputs**: load of access points, status of services and clients
- **Parameters manipulated**: association of a client to an access point, affecting number and types of clients per access point
- **Metric affected:** traffic load per access point, QoE per client
- **Dynamicity:** few seconds
- **Convergence:** one step
- **Correlation between iterations/decision making processes:** no
- **Explicit utility function:** no

4.3.3 Decentralized and adaptive network resource management

- **Topological scope**: core network segments
- **Inputs**: intra-domain link monitoring information
- **Parameters manipulated**: splitting rations for multi-topology routing
- **Metric affected:** intra-domain link utilization
- **Dynamicity:** every 10-15mins
- **Convergence:** iterative algorithm with a maximum 50 iterations; worst case execution time 5sec
- **Correlation between iterations/decision making processes:** feedback in terms of success/failure in shifting traffic during the previous iteration; however network state information is acquired once at the beginning of the algorithm so the actions of other NEMs do not affect the iterative process
- **Explicit utility function:** no

4.3.4 Cooperative remediation of vulnerabilities

- **Topological scope**: core network domains, VoIP infrastructure
- **Inputs**: configuration data of devices and services
- **Parameters manipulated**: configuration parameters and/or software versions of network devices
- **Metric affected:** traffic distribution slightly affected during remediation activities; configuration actions may affect traffic distribution depending on the nature of the configuration (e.g. altering device capabilities)
- **Dynamicity:** always "ON" NEM but taking actions only whenever a vulnerability is detected
- **Convergence:** one step
- **Correlation between iterations/decision making processes:** no
- **Explicit utility function:** no

4.3.5 Balancing CRE and ABS in 3GPP LTE HetNets

- **Topological scope**: LTE segments (macro cells and pico cells)
- **Inputs**: load information on pico and macro cells and status of CRE and ABS
- **Parameters manipulated**: control parameters of CRE (BIAS) and ABS (ABS ratio)
- **Metric affected:** traffic load per macro/pico cell, QoE per client
- **Dynamicity:** seconds to minutes
- **Convergence:** within 1sec
- **Correlation between iterations/decision making processes:** no
- **Explicit utility function:** no

4.3.6 ICIC and CCO coordination

- **Topological scope**: LTE segments (macro cells)
- **Inputs**: number of users associated with quality levels and user classes, ICIC and CCO objectives, weights for ICIC and CCO objectives, available resources of the target cell (bandwidth, maximum transmission power), users and eNodeB location, neighbouring cells context (location/transmission of eNodeBs, interference PRBs).
- **Parameters manipulated**: Physical Resource Blocks assignment, downlink transmission power
- **Metric affected:** cell interference, cell throughput, user throughput, spectral efficiency
- **Dynamicity:** semi-static (set by GOV)
- **Convergence:** one step
- **Correlation between iterations/decision making processes:** no
- **Explicit utility function:** yes, weighted version of ICIC and CCO related objective functions

4.3.7 SON Coordination strategies in LTE-advanced networks

LB and BRA Coordination (Load Balancing in HetNets with Relay Stations)

- **Topological scope**: LTE segments (macro cells with relays)
- **Inputs**: macro-cell load, nominal periodicity of the LB and BRA NEMs
- **Parameters manipulated**: relay pilot power (determining their coverage) and time multiplexing parameter (portion of resources allocated to the backhaul and to direct links)
- **Metric affected:** cell throughput, user throughput, outage rate, call blocking rate
- **Dynamicity:** LB in order of seconds, BRA in order of minutes
- **Convergence:** to fixed parameter values in order of minutes involving several steps
- **Correlation between iterations/decision making processes:** no, present action depends only on present load values (memoryless system)
- **Explicit utility function:** yes, each NEM uses and keeps a separate expression involving the load balancing objective

Coverage self-optimization

- **Topological scope**: LTE segments (macro cells)
- **Inputs**: coverage probability of neighbour base stations
- **Parameters manipulated**: traffic channels power
- **Metric affected:** cell throughput, user throughput, coverage probability
- **Dynamicity:** in order of seconds
- **Convergence:** to fixed parameter values
- **Correlation between iterations/decision making processes:** no, present action depends only on present load values (memoryless system)
- **Explicit utility function:** yes, each SON functionality uses expression involving the deviation of the neighbouring base stations coverage probability from a pre-defined target

4.3.8 Orchestration of resources and functions in edge networks

- **Topological scope**: edge networks (border between core domains - access domains)
- **Inputs**: virtual machine size, virtual resource status, edge node utility function
- **Parameters manipulated**: configuration parameters for virtual machine and network functions migration
- **Metric affected:** service downtime, migration time, connections throughput and delay
- **Dynamicity:** semi-static (set by GOV)
- **Convergence:** one step
- **Correlation between iterations/decision making processes:** no
- **Explicit utility function:** yes, weighted function of edge nodes utility functions

4.3.9 Coordinated link and node load balancing for virtualized evolved packet core

- **Topological scope**: LTE segments (mobile backhaul and core)
- **Inputs**: active bearers with S/PGW association per UE, current resource consumption (from GPN Virtualization Manager), current utilization of links (from switch/router component), utility functions in terms of load metrics
- **Parameters manipulated**: LSP/tunnel capacities, virtual machine resources and placement
- **Metric affected:** node and link load, traffic distribution, EPC function distribution
- **Dynamicity:** minutes to hours
- **Convergence:** iterative algorithm with no intermediate effects on the network; only final solution is provisioned if it differs from the previous
- **Correlation between iterations/decision making processes:** no
- **Explicit utility function:** yes, combination of load metrics

4.4 Implications

In this section we present a list of considerations, based on the behaviour and specification of the above described NEMs and instantiations of UMF COORD Core Mechanisms, that would need to be taken into account if they were to be concurrently deployed by an operator so as to address a network and service scenario.

While within the context of the project the above described NEMs and instantiations of UMF COORD Core Mechanisms were developed to address problems of specific Use Cases [58], we select to try to consider all of them together since scenarios and service setups might always arise that will require this, and the project use cases are in practice examples of situations that may need to be addressed by operators rather than strict placeholders for the project's developed NEMs and Core Mechanisms instantiations.

We do consider them though bound to the specific setup under which they were evaluated in Chapter 2 and Chapter 3 since the topological scope (environment) is a pre-requisite for identifying conflicts, dependencies and options for joint optimization.

The approach is based on cross-relating the various fields of information linked with each factor. Just as an example we summarize in Table 10 the fields for the NEMs of Section 4.3.1 and Section 4.3.3 respectively. Entries in **bold** are deemed as important for identifying potential conflicts, entries in *italics* are used when considering conflict resolution mechanisms and entries in ***bold italics*** are used for both.

Table 10. Example of cross-relating fields of information.

	Hybrid P2P selection for optimized user and network performance	Distributed and adaptive network resource management
Topological scope	**core network segments**	**core network segments**
Inputs	utilization/load of inter-domain links, domain business relationships, location of available peers	**intra-domain link monitoring information**
Parameters manipulated	the NEM does not explicitly alter a network parameter, but it generates a list of peers from which a specific content object is to be retrieved	splitting rations for multi-topology routing
Metric affected	**utilization/load of** inter-domain links and consequently the traffic load at the domain's ingress routers and **intra-domain links**	**intra-domain link utilization**
Dynamicity	***on a per content object request***	***every 10-15mins***
Convergence	***for each request there is one step***	***iterative algorithm with a maximum 50 iterations; worst case execution time 5sec***
Correlation between iterations	***no***	***feedback in terms of success/failure in shifting traffic during the previous iteration; however network state information is acquired once at the beginning of the algorithm so the actions of other NEMs do not affect the iterative process***
Explicit utility function	*no*	*no*

4.4.1 Hybrid P2P selection for optimized user and network performance

The NEM affects traffic going into ingress nodes of a core domain and is an always "ON" NEM. The changes it makes to the traffic are small and on a "per decision making" basis and successive decisions are not correlated. In addition it does not alter directly any network parameters. If content object requests are made by multi-access users (e.g. see the network setup for the evaluation of the Distribute Decision Engine), then DDE should be able to take these and their impact on the load of access points and the status of the video service into account and move the users to different access points. In all cases any changes to the incoming traffic volume and in its distribution among the core domain's ingress nodes, if reflected properly in network measurements, should not pose any problem to the Decentralised and Adaptive Network Resource Management NEM.

The risk of this NEM conflicting with other NEMs (and their grouping into Core Mechanisms) appears to be rather low and as such it may be allowed to operate unconstrained.

4.4.2 Distributed decision engine

The NEM affects the traffic load at access points but does not affect the route of traffic inside the core domain (see Figure 6 where both access points connect to the same border router; any routing changes take place after the border router and towards the access points). In addition successive decisions by the NEM are not correlated and there are no other NEMs operating in the wireless access domain. Therefore the NEM is

expected to react based only the current access point load and service conditions regardless of how the latter are affected from the source of the traffic until the users of the wireless access domain.

The risk of this NEM conflicting with other NEMs (and their grouping into Core Mechanisms) appears to be rather low and as such it may be allowed to operate unconstrained.

4.4.3 Distributed and adaptive resource management

This NEM only affects the internal flow of traffic within a core domain; the ingress and egress points are not affected meaning that any actions the NEM takes are confined -in terms of effects- within the considered core domain. The Hybrid P2P selection NEM can affect the distribution of traffic at ingresses, however these changes can be depicted by monitoring procedures and used at 15mins intervals by this traffic engineering NEM as a network snapshot. The decisions made by the NEM may affect the service status of the users connected through the wireless access points, but for DDE these changes are to be expected and DDE runs at a much shorter time scale (few seconds) which means that major changes at the service status -as a result of this NEM's operations- can take place only every 15mins. Even these though should not pose a problem to DDE.

It is expected therefore that this core TE NEM can operate unconstrained performing reconfigurations every 15mins, while DDE reacts to changes in the service status and access points load every few seconds with no additional implications. However, as it will be shown later, it may make sense to consider the timing of these 15mins intervals taking into account the timings of the Core Mechanism instantiations operating at edge networks and mobile core/backhaul.

4.4.4 Cooperative remediation of vulnerabilities

This NEM is fundamentally different from all the other NEMs and Core Mechanism instantiations in the sense that while always "ON", it only enforces actions whenever a certain abnormal situation arises. Therefore under "normal" conditions this NEM should not affect any other NEMs. If, however a vulnerability is detected and the remediation action involves changing a network device configuration which affects the operations of other NEMs (e.g. a patch applied to cure a vulnerability causes a router to have reduced capabilities or a router needs to be switched off altogether) then NEMs that in some way rely on that network device must become aware of this. Reliance can refer to NEMs residing or controlling this network device for their purposes or both; when residing on it (partially in case of distributed NEMs) then they should become aware of this and reorganise the communication and coordination between their distributed components, when controlling this device then the new device capabilities and allowed network parameter changes should be updated so that NEMs can reorganise their internal operations.

Based on the updated view of the network devices, COORD will need to check whether the possibility of a new conflict arised; in all cases one way to view this NEM is as non-conflicting with any NEM during normal conditions, while during problematic situations, in terms of vulnerability conditions, this NEM may have high priority in terms of actions (this can also be influenced by the operator). In such cases the rest of the NEMs will adjust their behaviour if needed and COORD itself will ensure that these updated behaviours do not lead to any new conflicts or, if they do, send a new NEM control policy [2] to the affected NEMs to guide their behaviour.

4.4.5 Balancing CRE and ABS in 3GPP LTE HetNets

This NEM operates in LTE deployments where pico cells co-exist with macro cells. Since all other NEMs and instantiations of COORD Core Mechanisms presented in this document are not designed to work with pico cells, there is no strong conflict regarding the pico cell part optimization of this NEM. In addition the flow of traffic to/from the mobile core and to/from the core segments should be reflected by monitoring procedures and since the NEM of Section 4.3.3 at the core domain operates in the order of 10-15mins, whereas the Core Mechanism instantiation of Section 4.3.9 at the mobile core/backhaul operates also in the order of minutes to hours, there is no obvious conflict between this NEM and the optimizations at core and mobile core/backhaul. (note that the same holds for all the NEMs/Core Mechanism instantiations at LTE segments; the core/mobile core optimizations are set to operate at fixed and longer scale intervals than the optimizations for the LTE segments and only rely on snapshots of the network conditions at the specific points that they are triggered; as a result they are not fundamentally affected by the changes made by optimizations at the LTE access segments, they rather view them as normal conditions to which they have to react accordingly)

If, however, some of the other LTE NEMs and Core Mechanism instantiations are operating in the same macro cell base stations or at nearby base stations, then conflicts can occur due to the same metrics being affected and loops being created. In such cases the possibility of including this NEM into a hierarchical optimization approach could be considered or a separation in time approach, where this NEM due to its fast convergence can be set to operate "in between" other NEMs that take longer to converge or are meant to be triggered more infrequently.

4.4.6 ICIC and CCO coordination

This instantiation of Core Mechanisms is designed to operate on macro cells and at rather long time scales (i.e. not in the order of seconds) and converges in one step. This makes it suitable for deployment in scenarios without high user mobility or changes in users' behaviour. Since it affects metrics that are indirectly or directly used by other NEMs, if it needs to be deployed in the same macro cells as other NEMs (e.g. the NEM performing macro and pico cell optimization), a separation in time strategy could be used where this Core Mechanism instantiation could set to operate at its nominal time scale with the NEMs with faster time scales operating "in between" and within the constraints set by it.

4.4.7 SON coordination strategies in LTE-advanced networks

LB and BRA Coordination (Load Balancing in HetNets with Relay Stations)

This instantiation of Core Mechanism is designed to operate in HetNets with relay stations and no other NEMs and/or Core Mechanism instantiations are designed to operate in that context. If there can be the case though that other optimization mechanisms need to operate in the same base stations (e.g. there are also pico cells in the proximity of the base stations) then due to possible conflicts one could consider the option of "incorporating" them all in a hierarchical optimization approach or clearly separate them in time; the latter would mean that once LB and BRA converge to fixed parameter values they would have to be "deactivated" (both LB and BRA NEMs are meant to be "always ON" even after they converge so as to be able to react to any deviations from their objectives). This decision would be up to the operator.

Coverage self-optimization

This NEM is designed to operate on macro cells at short time scales; the most obvious conflict may occur with ICIC and CCO Coordination Core Mechanism instantiation which is designed to operate in the same context. Both however affect the same control parameter (channel power) therefore one cannot run within the constraints of the other, in -for example- a time separation strategy. Concurrent optimization is not also possible due to the fundamentally different time scales. One possible solution would be to consider either of them alone for a specific part of the LTE segment depending on the user mobility and behaviour of the target location.

In case pico cells exist in the proximity of the base stations one may consider having this NEM and the NEM of Section 4.4.5 operating in parallel, since the joint macro-pico cell optimization converges very fast and this NEM converges slower but without being overly affected by changes in its input parameters since successive iterations are not correlated.

4.4.8 Orchestration of resources and functions in edge networks

This instantiation of Core Mechanism is designed to operate at the edge of core networks and provide functions and resources to manage requests coming from the users at the edge domains. If we consider the wireless access domain that DDE is managing in terms of user assignment to access points as one of these edge networks, the operations of this instantiation may be affecting the flow of traffic to (from) the border of the core domain from (to) the users of the wireless access domain and as a result the load of the access points and the status of the services monitored by DDE.

However, since DDE is intended to operate at much shorter time scales than this Core Mechanism instantiation and there is no correlation between DDE's successive decisions, there does not appear to be a conflict in their operation when considered together; changes incurred by this Core Mechanism instantiation will be reflected by monitoring procedures and DDE will react accordingly.

In a similar manner the flow of traffic towards/from the core may be affected; but the TE NEM operating at the core will see the changes reflected through monitoring procedures. However, since the core TE NEM is

triggered every 15mins it may make sense to have this trigger take place shortly after changes made by this Core Mechanism instantiation so that if there is any overload in the core segment due to changes in the edge networks, it can promptly be addressed by the core TE NEM.

4.4.9 Coordinated link and node load balancing for virtualized evolved packet core

This instantiation of Core Mechanism is intended to operate at the LTE mobile backhaul and core and optimize the paths and placement of functions. Flow of traffic from the mobile core/backhaul to core network domains may be affected, as is also the case for traffic flowing from the core network domains to the backhaul/mobile core.

Similar to the previous section, it may be useful to have this Core Mechanism instantiation coordinated in time with the core TE NEM so that any changes in traffic it incurs to/from the core domain can be addressed promptly by the core TE NEM. As such, the timings of the core TE NEM, of the Core Mechanism instantiation at edge networks and of the Core Mechanism instantiation at the mobile core/backhaul may all be considered together, having the core TE NEM react shortly after the optimizations at edge networks and at the mobile core/backhaul.

4.5 Discussion

As shown in this Chapter, there exists a number of factors that need to be taken into account when a number of NEMs and Core Mechanism instantiations are to be considered together with the prospect of further grouping them and guiding their behaviour. These factors inherently affect both the logic/need behind the grouping (e.g. NEMs operating in the same network segment are much more likely to conflict than NEMs operating in disjoint domains) and the feasibility of a Core Mechanism (e.g. one cannot apply a centralized multi-objective optimization type of Core Mechanism that relies on aggregating utility functions to group NEMs that do not use an explicit utility function even when optimizing in standalone mode). Deriving the dependencies and inter-relations in their behaviour, and consequently the proper way to streamline their behaviour, is not a trivial task.

In addition, the definition of conflict itself may be revisited depending on what an operator regards as a potentially problematic situation (possibly on a per-scenario basis); the same may hold for the importance an operator gives to conflict identification and resolution. For example, a set of NEMs may be considered as conflicting under a conservative operator A but may be considered as non-conflicting under a different operator B. As such the way to streamline the behaviour of a set of NEMs may vary depending on a scenario and an operator basis.

This Chapter attempted to highlight how the contributions presented in this deliverable relate with these factors and what considerations need to be taken into account. An attempt to generically reason about their inter-relations and how they translate in terms of grouping and control using the available COORD Core Mechanisms under the guidance of the COORD core block was given; apparently a more automated and also tuneable way to perform such reasoning is needed and this is an on-going task within WP2 for the needs of the deliverable D2.4 - Unified Management Framework (UMF) Specifications Release 3.

5 Conclusion

With the main focus of task T3.4 being to develop methods and strategies best-suited for cooperation between different entities, in this deliverable we presented the updates of the work conducted in task T3.4 in terms of NEMs that exhibit strong cooperation aspects in their operation (either due to their distributed nature or due to the kind of optimization they are performing, which attempts to balance multiple and contradicting objectives coming from different segments and/or services) and in terms of instantiations of UMF (COORD) Core Mechanisms that guide the interactions between independent NEMs.

As we showed, cooperation when it comes to individual NEMs' behaviour is something that can prove useful in a diverse set of network and service scenarios and can be implemented in many ways; each time to suit the particularities of the specific context as well as practical constraints in the operation of NEMs. Cooperation is beneficial in the context of both optimising NEMs (i.e. NEMs that are meant to be regularly taking decisions and enforcing network and service parameter configurations) and knowledge building NEMs.

With respect to UMF (COORD) Core Mechanisms we showed how instantiations of them can be applied to specific network scenarios and to specific NEMs. Examples of hierarchical optimization, synchronous control theory and centralized multi-objective optimization were considered, spanning almost the whole range of the project developed UMF (COORD) Core Mechanisms. The specific instantiations of COORD Core Mechanisms are an indication of their reusability and examples of successful mapping of mechanisms, from a specification point of view to an actual scenario/NEM-basis point of view.

Finally we attempted a holistic and reasoned approach towards integration through the UMF COORD core block. We defined how the contributions presented relate to a set of factors that would need to be considered so that conflicting situations on a per-NEM/Core Mechanism instantiation level can be avoided. As shown there exist multiple factors that need to be considered and deriving the dependencies and inter-relations in their behaviour, and consequently the proper way to streamline their behaviour, is not a trivial task. However, this approach, despite not being conclusive, it also showed that the COORD Core Mechanisms presented may be further reused to guide and control other NEMs, apart from the ones considered in the specific instantiations. This is very important since reusability of a mechanism is a major factor determining its actual incorporation in an operator's management system.

Future work on the NEMs and also with respect to the reusability of the Core Mechanisms is expected to be documented in Deliverable D3.9 "Handbook on optimization, learning, operation and cooperation methods" which by its nature will provide a benchmarking of mechanisms; this will provide practical guidelines to guide the actual selection of mechanisms to achieve cooperation both at an intra-NEM but also -and most important- at an inter-NEM level, subject to performance, network and service environment and other constraints.

The way to perform reasoning in terms of conflict identification and resolution by assignment of NEMs to the available COORD Core Mechanisms is also a related future activity; this will be performed in the context of WP2 for the needs of the deliverable D2.4 - Unified Management Framework (UMF) Specifications Release 3.

References

[1] UniverSelf D3.4 deliverable “Cooperation strategies and incentives”, December 2011.

[2] UniverSelf D2.2 deliverable “Unified Management Framework (UMF) Specifications Release 2”, October 2012.

[3] UniverSelf D3.5 deliverable “Adaptation and fine tuning of parameter optimization methods”, September 2012.

[4] UniverSelf D3.6 deliverable “Adaptation of learning and operation methods to specific needs of future networks and services”, September 2012.

[5] V. Aggarwal, A. Feldmann and C. Scheideler, “Can ISPs and P2P users cooperate for improved performance?” , ACM Computer Communications Review, Vol. 37, No. 3, pp. 29-40, July 2007

[6] The IETF ALTO WG, http://www.ietf.org/dyn/wg/charter/alto-charter.html

[7] S. Ren, E. Tan, T. Luo, S. Chen, L.Guo and X. Zhang, “TopBT: A topology-aware and infrastructure independent BitTorrent client”, IEEE INFOCOM 2010.

[8] Z. Dulinski, “Cost-driven Peer Rating Algorithm,” IEEE ICC 2011.

[9] P. Racz, S. Oechsner and F. Lehrieder, “BGP-Based Locality Promotion for P2P Applications,” IEEE ICCN 2010.

[10] X. Zhang, N. Wang and M. Howarth, “A Hybrid Peer Selection Scheme for Enhanced Network and Application Performances”, IEEE CCNC 2013.

[11] IEEE Std 802.21-2008, IEEE Standard for Local and Metropolitan Area Networks - Part 21: Media Independent Handover Services, IEEE, 2009.

[12] J. Sachs and M. Olsson, “Access network discovery and selection in the evolved 3GPP multi-access system architecture,” European Transactions on Telecommunucations, vol. 21, no. 6, pp. 544-557, 2010.

[13] J. Mäkelä, M. Luoto, T. Sutinen and K. Pentikousis, “Distributed information service architecture for overlapping multi-access networks,” Journal of Multimedia Tools and Applications, vol. 55, no. 2, pp. 289-306, 2011.

[14] M. Eisner, Ed., “XDR: External Data Representation Standard”, IETF Request for Comments 4506, 2006.

[15] C. Dannewitz, J. Golic, B. Ohlman and B. Ahlgren, “Secure Naming for a Network of Information,” in 13th IEEE Global Internet Symposium, San Diego, 2010.

[16] T. Rautio, M. Luoto, J. Mäkelä and P. Mannersalo, “Evaluation of autonomic load balancing in wireless multiaccess envinronment,” in IEEE WCNC, Shanghai, China, 2013.

[17] K. Pentikousis and T. Rautio, “A Multiaccess Network of Information,” in *WoWMoM*, Montreal, QC, Canada, 2010.

[18] D. Tuncer, M. Charalambides, G.Pavlou and N. Wang, “Towards decentralized and adaptive network resource management,” in Proc. of 7th IEEE/IFIP Conference on Network and Service Management (mini-CNSM), Paris, France, October 2011.

[19] P. Psenak et al., “Multi-topology (MT) routing in OSPF,” IETF RFC 4915, June 2007.

[20] M. Charalambides, G. Pavlou, P. Flegkas, N. Wang and D. Tuncer, “Managing the future Internet through intelligent in-network substrates,” IEEE Network, Special Issue: Managing an Autonomic Future Internet, Vol. 25, No. 6, Nov/Dec 2011.

[21] D. Tuncer, M. Charalambides, G. Pavlou and N. Wang, “DACORM: A coordinated, decentralized and adaptive network resource management scheme,” in Proc. of 13th IEEE/IFIP Network Operations and Management Symposium (NOMS), Hawaii, USA, April 2012.

[22] Y. Zhu, C. Dovrolis and M. Ammar, “Combing multihoming with overlay routing (or how to be a better ISP without owning a network),” in Proc. of the 2007 IEEE INFOCOM, Anchorage, Alaska, 2007.

[23] S. Fischer, N. Kammenhuber and A. Feldmann, “Replex: dynamic traffic engineering based on wardrop routing policies,” in Proc. of ACM CoNEXT conference (CoNEXT), Portugal, 2006.

[24] A. Kvalbein, C. Dovrolis and C. Muthu, “Multipath load-adaptive routing: putting the emphasis on robustness and simplicity,” in Proc. of 17th IEEE International Conference on Network Protocols (ICNP), USA, October 2009.

[25] M. Barrere, R. Badonnel, and O. Festor, “Towards the Assessment of Distributed Vulnerabilities in Autonomic Networks and Systems”, Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS’12), April 2012.

[26] M. Chiarini and A. Couch, “Dynamic Dependencies and Performance Improvement”, In Proceedings of the 22nd conference on Large Installation System Administration Conference, pages 9–21. USENIX, 2008.

[27] J. A. Wickboldt, L. A. Bianchin, and R. C. Lunardi, “Improving IT Change Management Processes with Automated Risk Assessment”, Proceedings of IEEE International Workshop on Distributed Systems: Operations and Management (DSOM’09), pages 71–84, 2009.

[28] J. A. Wang and M. Guo, “OVM: An Ontology for Vulnerability Management”, Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies (CSIIRW’09), pages 34:1–34:4, New York, NY, USA, 2009. ACM.

[29] The OVAL Language, http://oval.mitre.org/. Last visited on March 10, 2012.

[30] The XCCDF Language, Extensible Configuration Checklist Description Format, scap.nist.gov. Last visited on October 2012.

[31] M. Barrere, R. Badonnel, and O. Festor, “Collaborative Remediation of Configuration Vulnerabilities in Autonomic Networks and Services”, Proceedings of the IEEE/IFIP Conference on Network and Service Management (CNSM’12), October 2012.

[32] S. Parkval, E. Dahlman, G. Jöngren, S. Landström and L. Lindbom, “Heterogeneous network deployments in LTE”, Ericsson Review, No 2, 2011.

[33] 3GPP TS36.331 Radio Resource Control (RRC)

[34] 3GPP TS36.443 X2 application protocol (X2AP)

[35] 3GPP TR36.814 Further advancements for E-UTRA physical layer aspects

[36] R. Combes, Z. Altman and E. Altman, “Self-organizing relays: dimensioning, self-optimization and learning”, IEEE Transactions on Network Management, TNSM, Dec. 2012.

[37] R. Combes, Z. Altman and E. Altman, “Coordination of autonomic functionalities in communications networks”, submitted to WiOpt, Torino, Tsukuba Science City, Japan 2013. (Also available at http://arxiv.org/abs/1209.1236).

[38] 3GPP, “Evolved Universal Terrestrial Radio Access Network (EUTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions,” 3rd Generation Partnership Project (3GPP), TR 36.902, March 2011. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36902.htm

[39] P. Vlacheas, E. Thomatos, K. Tsagkaris and P. Demestichas, “Autonomic Downlink Inter-Cell Interference Coordination in LTE Self-Organizing Networks", In Proc. 7th International Conference on Network and Service Management (CNSM 2011) – Poster Sessions, Paris, France, 2011.

[40] Tao Cai, G. Koudouridis, C. Qvarfordt, J. Johansson and P. LeggV, “Coverage and Capacity Optimization in E-UTRAN Based on Central Coordination and Distributed Gibbs Sampling”, In Proc. IEEE 71st Vehicular Technology Conference (VTC 2010-Spring), Taipei, 2010, pp. 1 – 5.

[41] R. Razavi, S. Klein and H. Claussen, “Self-Optimization of Capacity and Coverage in LTE Networks Using a Fuzzy Reinforcement Learning Approach”, In Proc. IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2010, pp. 1865 – 1870.

[42] SOCRATES project website [Online]. Available: http://www.fp7-socrates.org/

[43] T. Jansen, M. Amirijoo, U. Türke, L. Jorguseski, K. Zetterberg, R. Nascimento, L.C. Schmelz, J. Turk and I. Balan, “Embedding multiple self-organisation functionalities in future radio access networks”, In Proc. IEEE 69st Vehicular Technology Conference (VTC 2009-Spring), Barcelona, 2009, pp. 1 – 5.

[44] L.C. Schmelz, J.L. van den Berg, R. Litjens, K. Zetterberg, M. Amirijoo, K. Spaey, I. Balan, N. Scully and S. Stefanski, “Self-organisation in wireless networks - use cases and their interrelations”, In Proc. Wireless World Research Forum Meeting 22, Paris, 2009.

[45] L.C. Schmelz, M. Amirijoo, A. Eisenblätter, R. Litjens, M. Neuland and J. Turk, “A coordination framework for self-organisation in LTE networks”, In Proc. 2011 IFIP/IEEE International Symposium on Integrated Network Management (IM), Dublin, 2011, pp. 193 – 200.

[46] T. Bandh, H. Sanneck and R. Romeikat, “An Experimental System for SON Function Coordination”, In Proc. IEEE 73rd Vehicular Technology Conference (VTC 2011-Spring), 2011, pp. 1 – 2.

[47] F. Chen and G. Tao, “A Novel MCS Selection Criterion for Supporting AMC in LTE System”, In Proc. International Conference on Computer Application and System Modeling (ICCASM 2010), 2010, pp. 598 – 603.

[48] T. Bonald and A. Proutiere, “Wireless downlink data channels: User performance and cell dimensioning”, ACM Mobicom 2003.

[49] R. Combes, Z. Altman and E. Altman, “Self-organization in wireless network: a flow-level perspective”, IEEE INFOCOM 2012, Orlando, US, March 2012.

[50] J. B. Rosen, “Existence and Uniqueness of Equilibrium Points for Concave N-Person Games”, Econometrica, vol. 33, no. 3, pp. 520–534, 1965.

[51] T. Kelly, “Utility-directed allocation”, In First Workshop on Algorithms and Architectures for Self-Managing Systems, 2003.

[52] A. Byde, M. Salle, and C. Bartolini, “Market-based resource allocation for utility data centers”, Technical Report HPL-2003-188, HP Laboratories Bristol, 2003.

[53] D. Gelernter, “Generative Communication in Linda”, Yale University, 1983.

[54] P.J.M. van Laarhoven and E.H.L. Aarts, “Simulated Annealing: Theory and Applications”, Kluwer Academic Publisher, 1987.

[55] A. A. Zhigljavsky, “Theory of Global Random Search”, Kluwer Academic Publishers, 1991.

[56] F.Z. Yousaf, J. Lessmann, P. Loureiro, S. Schmid, “SoftEPC – Dynamic Instantiation of Mobile Core Network Entities for Efficient Resource Utilization”, IEEE ICC 2013

[57] L. Schmelz, “SON Coordinator – SOCRATES approach”, presentation at RAS cluster meeting, October 2010, (http://www.fp7-socrates.org/files/Presentations/SOCRATES_2010_RAS%20cluster%20presentation.pdf)

[58] UniverSelf D4.2 deliverable “Synthesis of Use Case Requirements – Release 2”, April 2012.

Abbreviations

3GPP	3rd Generation Partnership Project
ABS	Almost Blank Subframe
ANDSF	Access Network Discovery and Selection Function
AP	Access Point
AS	Autonomous System
BER	Bit Error Rate
BCR	Block Call Rate
BNG	Broadband Network Gateway
BRA	Backhaul Resource Allocation
CAPEX	Capital Expenditures
CCO	Coverage and Capacity Optimization
CDF	Cumulative Distribution Function
CRE	Cell Range Extension
DDE	Distributed Decision Engine
DE	Decision Entity
DNS	Domain Name Server
DOVAL	Distributed OVAL
DoW	Description of Work
DPI	Deep Packet Inspection
DT	Distributed Treatment
DV	Distributed Vulnerability
DXCCDF	Distributed XCCDF
eNodeB	Evolved NodeB
EPC	Evolved Packet Core
FTP	File Transfer Protocol
GGSN	Getway GPRS support Node
GPN	General Purpose Node
GW	Gateway
HetNet	Heterogeneous Network
HLR	Home Location Register
HSS	Home Subscriber Server
ICIC	Inter-Cell Interference Coordination
IDS	Intrusion Detection system
IE	Information Element
IMS	IP Multimedia Subsystem
INO	In-Network Overlay
IoT	Internet of Things
IP	Internet Protocol
IPsec	Internet Protocol Security
IPTV	Internet Protocol Television
ISP	Internet Service Provider
KPI	Key Performance Indicator
LB	Load Balancing
LSP	Label Switched Path
LTE	Long Term Evolution

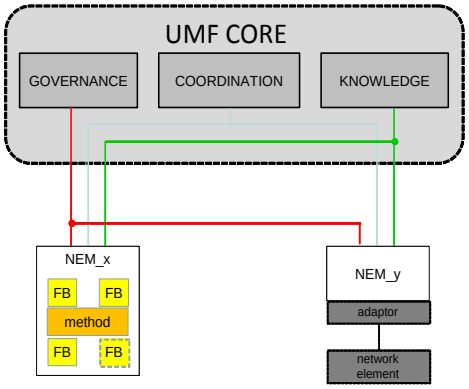
LTE-A	LTE Advanced
MIH	Media Independent Handover
MLB	Mobility Load Balancing
MME	Mobility Management Entity
MPLS	Multi Protocol Label Switching
MRO	Mobility Robustness
NAT	Network Address Translation
NEM	Network Empowerment Mechanism
NFV	Network Functions Virtualization
NMS	Network Management System
NSC	Network Stability and control
NSGA	Non-dominated Sorting Genetic algorithm
ODE	Ordinary Differential Equation
OFDM	Orthogonal Frequency-division Multiplexing
OFDMA	Orthogonal Frequency-Division Multiple Access
OPEX	Operational Expenditures
OVAL	Open Vulnerability and Assessment Language
PDCCH	Physical Downlink Control Channel
PCRF	Policy and Charging Rules Function
PGW (also PDN-GW)	Packet Data Network Gateway
PRB	Physical Resource Block
P2P	Peer-to-Peer
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RNTP	Relative Narrowband Transmission Power
SA	Simulated Annealing
SBC	Session Border Controller
SDN	Software Defined Network
SE	Selected Entity
SGW	Serving Gateway
SINR	Signal to Interference-plus-Noise Ratio
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SON	Self Organized Networks
SSL	Secure Sockets Layer
TDF	Traffic Detection Function
TCP	Transmission Control Protocol
TE	Traffic Engineering
TOF	Traffic Offload Function
UC	Use case
UE	User Equipment
UMF	Unified Management Framework
VoD	Video on Demand
VPN	Virtual Private Network
WAN	Wide Area Network

XCCDF	eXtensible Configuration Checklist Description Format
XDR	External Data Representation
XML	eXtensible Markup Language

Definitions

Network Empowerment Mechanism (NEM) – *a functional grouping of objective(s), context and method(s) where "method" is a general procedure for solving a problem. A NEM is (a priori) implemented as a piece of software that can be deployed in a network to enhance or simplify its control and management (e.g. take over some operations). An intrinsic capability of a NEM is to be deployable and interoperable in a UMF context (in a UMF-compliant network).*

Unified Management Framework (UMF) – *A framework that will help produce the unification, governance, and "plug and play" of autonomic networking solutions within existing and future management ecosystems. The objective of the UMF is to facilitate the seamless and trustworthy deployment of NEMs. The UMF has three core blocks that are used by the NEMs to achieve this, as shown in the figure below.*



Governance block (GOV) – *A core UMF block that aims to give a human operator a mechanism for controlling the network from a high level business point of view, that is, without the need of having deep technical knowledge of the network.*

Knowledge block (KNOW) – *An infrastructure that uses and/or manipulates information and knowledge, including information/knowledge flow optimization within the network.*

Coordination block (COORD) – *A core UMF block that aims to ensure the proper sequence in triggering of NEMs and the conditions under which they will be invoked (i.e. produce their output), taking into account operator service and scenario requirements and at the same time the needs for conflict avoidance, stability control and joint optimization through the corresponding functions.*

UMF core mechanism – *Any functionality residing within any of the UMF core blocks*

Use case – *A descriptor of a set of precise problems to be solved. It describes steps and actions between stakeholders and/or actors and a system, which leads the user towards an added value or a useful goal. A use case describes what the system shall do for the actor and/or stakeholder to achieve a particular goal. Use-cases are a system modelling technique that helps developers determine which features to implement and how to gracefully resolve errors.*