



SEVENTH FRAMEWORK PROGRAMME

THEME ICT-1-1.1

“Network of the future”

Project acronym: **EFIPSANS**

Project full title: **E**xposing the **F**eatures in **I**P version **S**ix protocols that can be exploited/extended for the purposes of designing/building **A**utonomic **N**etworks and **S**ervices

Proposal/Contract no.: **INFSO-ICT-215549**

EFIPSANS Integration Framework

Project Document Number: EFIPSANS/D5.1 CO¹ v7 (official version)

Project Document Date: 30/04/2010

Workpackage Contributing to the Project Document: WP5, additional support from WP2, WP3 and WP4

Deliverable Type and Security: P,D/PU

Editors: Peter Benko (ETH), Vassilios Kaldanis (VELTI)

Co-Editor: Domonkos Asztalos (ETH)

Abstract: The deliverable will document the methodology for integrating autonomic scenarios for heterogeneous networking environments. It serves as an initial specification document concerning the final demo and provides a comprehensive description of the testbeds that will be used for the execution of the demonstrations. The scenario descriptions serve as the base material for the specification of the Integrated Framework, and based on it the scenario development and selection process for the Integrated Demonstrator is provided. Also a validation methodology is prescribed for validating the functionality, autonomic behaviors and features of the proposed approaches and architectures.

Keywords: EFIPSANS, scenarios, demonstrator, integrated testbed

¹ Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Project Number	INFSO-ICT-215549
Project Name	Exposing the Features in IP version Six protocols that can be exploited/extended for the purposes of designing/building Autonomic Networks and Services
Document Number	INFSO-ICT-215549/WP5/D.5.1
Document Title	EFIPSANS Integration Framework
Workpackage	WP5, additional support from WP2, WP3 and WP4
Editor Co-Editor	Peter Benko (ETH) Domonkos Asztalos (ETH) Vassilios Kaldanis (VELTI)
Authors	Peter Benko (ETH) Domonkos Asztalos (ETH) Csaba Simon (BME) Vassilios Kaldanis (VELTI) Contributors from WP2, WP3 and WP4.
Reviewers	Symeon Papavassiliou Ranganai Chaparadza Tony Jokikyyny
Contractual delivery date	30th April 2010
Delivery Date	30th April 2010
Version	7.0/ 30th April 2010

Copyright

Material on these pages is copyright EFIPSANS Project except where references to other original sources are made. It may be downloaded and printed to use in classrooms, lectures, research projects, industry reference, etc or cited in other documents with due credit to the EFIPSANS project and the authors; but not otherwise copied, altered in any way or transmitted to others. Web locations are for convenience of users and do not constitute any endorsement or authorisation by EFIPSANS Project.

Contact authors: Peter Benko (peter.benko@ericsson.com), Domonkos Asztalos (domonkos.asztalos@ericsson.com), Vassilios Kaldanis (vkaldanis@veltiservices.com)

Executive summary

The EFIPSANS Integration Framework is designed for demonstrating a substantial selection of essential autonomic behaviors whose implementations are based on the GANA principles and on the IPv6 protocol extensions proposed and developed in the frame of the project. The demonstrated autonomic behaviors include the following network functionalities: Routing and Forwarding, Auto-Discovery and Auto-Configuration, Mobility Management, QoS Management, Resilience and Survivability, Self-Monitoring and Fault-Management.

Those autonomic behaviors were embedded into selected scenarios assuming heterogeneous network environment and demonstrating the benefits of the EFIPSANS concept for the key players, like network operator, Internet service provider and end user. A Scenario Template was defined for driving the uniform documentation of the scenarios. Twelve scenarios were invented altogether. A scenario categorization process was pursued to reach the decision which scenarios will form the core of the integrated demonstrator, which scenarios will be used as satellite demonstrators connected with the core demonstrator and finally which scenarios will be implemented as standalone demonstrators.

The heterogeneous network environment is provided through the implementation of several testbeds located at different partners and connected with IP tunnels. The fixed network environment testbeds are used most extensively, while a wireless (IEEE 802.11) environment is supported as well. Beyond those physical testbeds there is an emulated integrated WCDMA/WLAN radio access network environment used in one of the standalone demonstrators.

Among the core components of the physical testbeds are the routers. In the core integrated testbed, the project uses two types of routers with the aim of extending them with selected GANA components and IPv6 protocol extensions: Quagga soft router and the Ericsson SmartEdge router. The interoperability of those two router types will be demonstrated. The central element of the information dissemination within the network will be the ONIX.

Among others, the following IPv6 protocol extensions will be demonstrated: DHCPv6++, ICMPv6++, NDv6++ and the IPv6 header extension for QoS. A substantial set of Decision Elements will be implemented in the frame of the demonstrators.

A video streaming service will be used in the integrated demonstrator. The final configuration will have a handful of video servers and hosts representing the customers.

The execution and the verification of the demonstrators will be supported with a visualization tool providing a GUI to follow the changes in the network state.

In WP5, the verification and validation activity will concentrate on the functional correctness, as well as the autonomic behaviors and operational features of the proposed approaches and architectures

Table of Contents

<i>Abbreviations</i>	6
<i>List of Figures</i>	8
<i>List of Tables</i>	10
1 Introduction	11
1.1 Scope of the Deliverable	11
1.2 Structure of the Document	12
2 Methodology	13
2.1 Introduction	13
2.2 Scenario Template	14
2.2.1 Part A of the Scenario Description Template	14
2.2.2 Part B of the Scenario Description Template.....	16
2.3 Scenarios Overview and Evaluation	17
2.4 Testbed Integration Methodology	19
2.4.1 The Integrated Testbed	20
2.4.2 The Satellite Testbeds.....	20
2.4.3 The Standalone Testbeds	20
2.5 Validation methodology	21
2.5.1 Concepts to validate.....	21
2.5.2 Components to Validate.....	23
3 Description of the Testbeds	24
3.1 Description of the Integrated Testbed	24
3.1.1 Architecture	24
3.1.2 Integration of the testbed components	25
3.1.2.1 Platform integration	25
3.1.2.2 IPv6 integration.....	26
3.1.2.3 Functional integration and validation.....	26
3.1.2.4 Integrated visualization	26
3.1.2.5 Illustrations of other Visualization Methods to be considered for GANA.....	29
3.2 Description of the Satellite Testbeds	33
3.2.1 BUPT testbed.....	33
3.2.2 MANET testbed.....	35
3.3 Description of the Stand-alone Testbeds	37
3.3.1 Autonomic Mobility and QoS Management	37
3.3.2 WARF testbed	40
3.3.3 GRNET testbed.....	41
4 Specification of the Final Demonstration	42
4.1 List of demonstrated protocol extensions	42
4.2 List of demonstrated Components	42
4.3 Overview of the final demo	46
5 Conclusion and Next Steps	47
<i>References</i>	48
<i>Appendix A– EFIPSANS Scenarios</i>	49
A-1.1 Basic Autonomic Networking Scenarios (WP2)	49

A-1.1.1 Auto-Discovery and Auto-Configuration Scenarios.....	50
A-1.1.1.1 Self-Configuration of Routers using Routing Profiles in Fixed Network Environment.....	50
A-1.1.1.2 Auto-Collaboration for Optimal Network Resource Utilization in a Fixed Networks	56
A-1.1.2 Auto- and Self-Configuration Scenarios.....	67
A-1.1.2.1 Auto-Configuration/Self-Configuration of Addresses in a SOHO Network	67
A-1.1.2.2 Auto-Configuration/Self-Configuration of Addresses in MARSIAN Platform	71
A-1.1.2.3 Auto-Configuration of Radio Channels in 802.11 networks	79
A-1.1.3 Autonomic Routing Scenarios	83
A-1.1.3.1 Autonomic multipath routing in 802.11 Mesh Networks	83
A-1.1.3.2 Autonomic Routing and Self-Adaptation driven by Risk-Level Assessment in Fixed Network Environments.....	88
A-1.2 Advanced Autonomic Networking Scenarios (WP3).....	99
A-1.2.1. Autonomic Mobility and QoS Management Scenario	99
A-1.2.2 Autonomic QoS Management in Wired Network.....	106
A-1.3 Autonomic Network Management Scenarios (WP4).....	111
A-1.3.1 Autonomic Peer-To-Peer (p2p) Network Monitoring Scenario.....	111
A-1.3.2 Network Monitoring and QoS Management Scenario.....	119
A-1.3.3 Autonomic Fault-Management Scenario	124

ABBREVIATIONS

AB	Autonomic Behavior
ADA	Autonomic DHCP Architecture
AP	Access Point
BR	Border Router
BS	Base Station
CP	Content Provider
DE	Decision Element
DHCP	Dynamic Host Configuration Protocol
GAN	Generic Autonomic Network Architecture
GUI	Graphical User Interface
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
MANET	Mobile Ad hoc NETWORK
MARSIAN	Platform for fault Management, Auto-configuration, Resilience and Survivability In Mobile Ad hoc Networks
ME	Managed Entity
ND	Neighbor Discovery
ONIX	Overlay Network for Information eXchange
OSFP	Open Shortest Path First
PDA	Personal Digital Assistant
QoS	Quality of Service
SOHO	Small Office/Home Office
WARF	Wireless and Autonomic Routing Framework

W-CDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network

LIST OF FIGURES

Figure 1: WP5 relationship with the other WPs	13
Figure 2: EFIPSANS Scenarios and Testbeds	19
Figure 3: EFIPSANS Integration Environment	23
Figure 4: Integrated Demonstrator Architecture	24
Figure 5: Visualization framework	27
Figure 6: Visualizer tool – Graphical User Interface	28
Figure 7: Visualization of DE interactions involved in the autonomic QoS and Mobility Management Scenario.....	30
Figure 8: Advanced visualization of DE interactions in the autonomic QoS and Mobility Management Scenario.....	31
Figure 9: <i>Visualization of synchronization of actions in autonomic Fault-Management and Resilience and Survivability within a node</i>	32
Figure 10: BUPT testbed topology	33
Figure 11: TARC-PL testbed: physical network topology from wired Ethernet interface perspective and testbed setup overview	35
Figure 12: TARC-PL testbed: physical network topology from wired Ethernet interface perspective and the resulting multi-hop MANET	36
Figure 13: The scenario's environment and key actors	37
Figure 14: Mapping our Wireless Emulator to a Real Architecture (3GPP/LTE)	38
Figure 15: Emulated Wireless Environment	39
Figure 16: Example GUI for the scenario	40
Figure 17: Autonomic Provision of Services	41
Figure 18: Aggregation of Capabilities of a Node by the NODE_MAIN_DE	54
Figure 19: Initial configuration	60
Figure 20: A New Router Is Added Into the Network	61
Figure 21: Inter-domain Resource Discovery	63
Figure 22: Sequence diagram for using the Ingress_Egress_Choice_Metric	64
Figure 23: GANA architecture overview for MARSIAN platform	76
Figure 24: An example situation covered by the proposed scenario	77
Figure 25: GANA architecture overview for WARF platform	86
Figure 26: Temperature of BR1 reaches threshold limit – Moderate risk	93
Figure 27: Temperature of BR1 crosses threshold limit – High Risk.....	95
Figure 28: Temperature of BR1 reaches threshold limit – Maximum Risk.....	97
Figure 29: Setting up P2P overlay network though P2P_ME.....	117
Figure 30: Neighbor Discovery through Vis_ME.....	117
Figure 31: Performance Monitoring through TM_ME	118
Figure 32: Admission control through AC_ME	118
Figure 33: QoS Violation Scenario.....	120
Figure 34: Admission Control Scenario.....	122

Figure 35: Testbed topology	123
Figure 36: Possible causes for black holes.....	127
Figure 37: An Abstract Black Hole Scenario Setup.....	128
Figure 38: The Autonomic Fault-Management and Reactive Resilience in EFIPSANS.....	129
Figure 39: A testbed realizing the IP black hole Scenario	130
Figure 40: R2-R4 link is Down now the traffic is rerouted over a path with a smaller PMTU.....	131

LIST OF TABLES

Table 1: Scenarios defined by EFIPSANS	17
Table 2: List of NETWORK LEVEL DEs	42
Table 3: List of NODE LEVEL DEs	43
Table 4: List of FUNCTION LEVEL DEs	43
Table 5: List of the PROTOCOL LEVEL DEs	44
Table 6: List of the Managed Entities (MEs).....	45
Table 7: Metrics values over time.....	65

1 INTRODUCTION

This deliverable analyzes in detail the EFIPSANS demonstration framework comprising mainly of the scenario elaboration and use case definition, the integration methodology and testbed related specification, and finally validation and assessment. The ultimate goal of WP5 is to demonstrate various autonomic behaviors and features along with the corresponding EFIPSANS extensions to IPv6 protocols providing contents that will be further developed and disseminated to IETF.

Furthermore, this deliverable provides the high level specification of the EFIPSANS Integration Framework. This high level specification of the Integration Framework will be followed by two deliverables (D-5.2 and D-5.3) describing the achieved results and respective demonstrators. The scenario descriptions serve as the base material for the specification of the Integrated Framework. For reader's convenience and for completeness purposes all scenarios from WP2, WP3 and WP4 have been specified and can be found in the Appendix of this document.

The actual specification of the EFIPSANS Integration Framework is based on several other project deliverables produced by the technical work packages (WP1-WP4). The most important ones are the following:

- D-1.5 Autonomic Behaviors Specifications (ABs) for Selected Diverse Networking Environments [1]
- D-1.5/SD Specification and Description Tables for Decision Elements [2]
- D-2.3 Initial Draft Specification of the required extensions to IPv6 protocols [3]
- D-2.4 Selected feature combination scenarios for Demo [4] – it describes the scenarios developed by WP2.

The role of WP5 is to specify and develop the framework and integration environment for the testbed implementation and the corresponding scenarios' realization. Initially WP5 performed collection of all scenarios and their requirements as determined by potential trials/demos/scenarios. As a result a set of testbeds, namely integrated testbed, satellite testbeds and standalone testing environments, have been specified. The integrated testbed environment is used so that more complex scenarios are integrated into some trial cases requiring the integration and deployment of all software components resulting from different frameworks designed within EFIPSANS.

1.1 Scope of the Deliverable

The main goal of the development of the EFIPSANS Integration Framework is to showcase the results of the EFIPSANS developed Frameworks, related mainly to Autonomic Routing and Forwarding, Auto-Discovery and Auto-Configuration, Mobility and Autonomicity, QoS and Autonomicity, Resilience and Survivability, Self-Monitoring and Autonomic Fault-Management. This means that the main EFIPSANS integrated testbed with all its networking environments will consist of nodes that implement software components, protocols and algorithms related to the seven main categories of autonomic functions, all contributed collectively by WP2, WP3 and WP4. The deliverable aims at documenting the methodology for integrating autonomic scenarios for heterogeneous networking environments. It serves as an initial specification document concerning the final demo and provides a description of the testbeds that will be

used for the execution of the demonstrations. Also a validation methodology is provided for validating the functionality, autonomic behaviors and features of the proposed approaches.

The fulfillment of the main goal will result in compliance to both the completeness and the heterogeneity requirements, where the first is about to demonstrate autonomic behaviors in all the seven functional areas listed above, the second is to demonstrate the autonomic behaviors in several network environments.

Another important goal of the development of the EFIPSANS Integration Framework is to showcase the benefits of the developed protocol extensions, software architecture, algorithms for the key players of the networking domain, such as network operators, content providers and end-users.

In summary, the main goals of WP5 are:

- To demonstrate the role and significance of the identified exploitable existing IPv6 features.
- To set up the trial to test EFIPSANS extensions to IPv6 protocols
- To present/demonstrate newly introduced algorithms, paradigms, network architectural extensions, and novel network management functions developed in EFIPSANS
- To evaluate the selected/developed autonomic behaviour scenarios in testbed field trials.
- To test EFIPSANS ideas and to demonstrate the achieved results for industrial applications.
- To develop a field trial system that can be used for prototyping and demonstrating the EFIPSANS architecture.
- To develop and demonstrate a real case application and services scenario that tests and evaluates the interaction of autonomic features in core and wireless networks.

1.2 Structure of the Document

The structure of the deliverable is as follows: Chapter 2 describes the scenario development and selection process for the Integrated Demonstrator. In Chapter 3, a description of the core integrated testbed, the satellite testbeds and the standalone testbeds is given. Some characteristics of the testbeds used for the implementation and execution of the demonstrators are described in several deliverables, however the current document provides a comprehensive description of all of them. Chapter 4 describes the structural components and a draft story-line for the Integrated Demonstrator. The Appendix contains the scenario descriptions.

2 METHODOLOGY

2.1 Introduction

This section presents the methodology adopted within WP5 to successfully accomplish the development of the EFIPSANS demonstration environment. It comprises the following parts:

- Demonstration Scenario elaboration
- Testbed Integration Methodology
- Integrated Testbed Definition (Architecture and Specification)
- Additional Satellite and Stand Alone testing environments

WP5 fulfills the requirements of EFIPANS integration through its successful interaction with the project's individual technical WPs and other important activities within the project (e.g. business estimation, exploitation, standardization, etc) as seen in Figure 1 below. Scenarios elaboration has been performed by the respective WPs under WP5 guidance using a specific template (see section 2.2). The objective was to highlight all scenarios different aspects and features not only on the technical part but also on the business one, fully addressing the expected impacts (involved actors, players, end users, etc) and outcomes of their actual deployment.

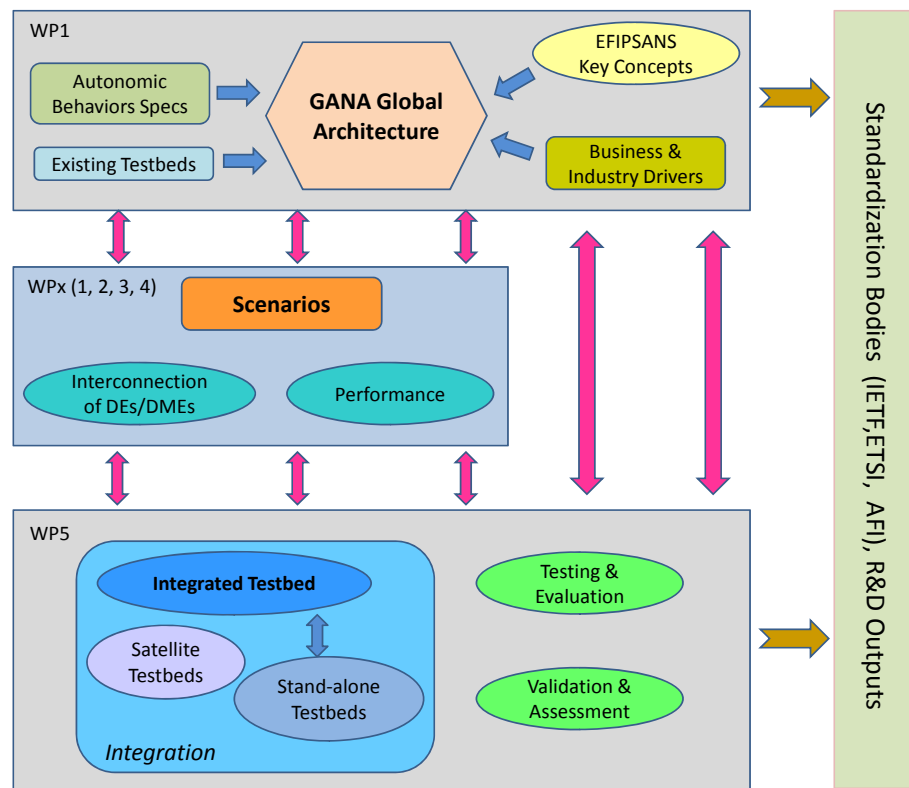


Figure 1: WP5 relationship with the other WPs

2.2 Scenario Template

Among the main targets and tasks of WP5 was to collect, consolidate and integrate all project scenarios targeted for demonstration. The outcome of this activity is a scenario manual (see the Appendix) that:

- Serves as a base document to address integration requirements
- Provides support for prototyping activities through addressing low-level technical information. It also provides consolidated information about system specifications and requirements, testbed features and parameters.
- Can be used to perform a comparative analysis with third party approaches
- Encourages the usage of a common language among partners in the scenario and use case elaboration in terms of demonstrated autonomic behaviors, metrics of their evaluation and assessment.

Each Scenario is described following a Template that is described below. It consists mainly of two parts: a High-Level-Description part and a Detailed Technical Description part. Almost all early scenarios that have been initially documented during the preparatory WP5 phase (1st project year) as part of other activities (e.g. requirement analysis, architecture instantiation) are now described according to the template. Key issues for focus and further improvement in the next phase of scenarios implementation have been identified and included in this template.

2.2.1 Part A of the Scenario Description Template

The part A of the scenario template used in EFISPANS includes the following parts:

Scenario Name

< Name >

(e.g. Auto-Configuration of Routers using Routing Profiles in a Fixed Network Environment)

Brief introduction

< A short abstract description of the Scenario that is being described >

High-Level Description of the Scenario

< high level information of the Scenario provided in the following tabular format >

Scenario Descriptive Name <NAME> (e.g. Auto-Configuration of Routers using Routing Profiles in a Fixed Network Environment)	
The Story-line	<Abstract description of the scenario details minimizing technical information, provide clear view of the number of steps states or scenes/phases that the service is realized within, name the involved actors (e.g. end-user, operator), details about the environment the service activity takes place (e.g. the user detects more than one access points/networks, he is equipped with a multimode device), some flavour of problematic

	behaviour that the scenario aims to bring to attention (e.g. the user experiences poor service quality and looks manually for a solution), etc>	
Short description of the scenario	<This part provides a more technical but brief description providing specific details (e.g.) about underlying systems involved, device capabilities, expected or identified functionality or operation applied (e.g. resource discovery)>	
Scenario Scenes	<This part provides a split of the scenario description into particular scenes with particular technical importance in relation to the phases constituting the Scenario (e.g. network selection is in progress....)>	
Current problems / limitations with current practices and/or current technology	<A detailed summary description of the problems/limitations with the current network management practices and/or current technology with respect to the scenario being described> [Wherever relevant, we talk about either “current practices” and/or “current technology that come with the devices/systems”. The Scenario then reflects that we solve those issues]	
Network Environment	e.g. Fixed/Wired/Hybrid/Overlay/Converged/Heterogeneous/Pervasive	
Self-* Functionalities introduced	Self-*Functionality	Problems/Limitations it addresses [Wherever relevant, we talk about either “current practices” and/or “current technology that come with the devices/systems”. The Scenario then reflects that we solve those issues]
	<i>Self-X1 (name)</i>	1. <u>Current Practices</u> : Current Practice P1 : <Here we explain the current practices applied today in this specific domain, and what are the problems/limitations thereof> Current Practice Pn : <other / similar practice into attention> 2. <u>Current Technology</u> : Current Technology <T1> : <What are the problems/limitations that come with current technology of devices/systems? Expectations on how underlying technology will be impacted from Self-X functionality> Current Technology <Tn> : Same for Tn
	<i>Self-Xn (name)</i>	1. <u>Current Practices</u> : Current Practice Pm Current Practice Pn 2. <u>Current Technology</u> :

		Current Technology <Tm> Current Technology <Tn>
Self-X1 – What it solves and the benefits	<What problem the self-X functionality solves in real terms and parameters (e.g. avoid or minimize the manual configuration effort for the device/router), and the expected benefits (long/short term ones) from the perspectives of the advancement of the technologies impacted by the Self-X functionality and the Service/applications performance (in)directly relying on the functionality>	
Self-Xn – What it solves and the benefits		
System(s) Involved	<Systems involved in the overall scenario (e.g. Core/Edge/Access routers)>	
Key players that benefit	Actor/Player	Benefits (with Rationale)
	Operator	How the operator is involved
	Manufacturer	How the manufacturer is involved
	End User	How the end-user is involved

2.2.2 Part B of the Scenario Description Template

Detailed Technical Description of the Scenario

<This part provides a full description of the scenario (with Decision Elements (DEs), interconnections among them, functionality instantiation, etc)>

2.3 Scenarios Overview and Evaluation

Table 1 below shows the list of scenarios defined within EFIPSANS per different activity within each work package, along with the collaborating partners. The full scenario template-based descriptions are provided in the Appendix-A of the present deliverable.

Table 1: Scenarios defined by EFIPSANS

	Scenario Description	Collaborating Partners
Auto-Configuration / Auto-Discovery		
1	Auto-Discovery and Auto-Configuration (Self-Configuration) of Routers using Routing Profiles in Fixed Network Environment	Fraunhofer FOKUS,ALBLF, Ericsson ETH, Ericsson LMF
2	Auto-Collaboration for Optimal Network Resource Utilization in Fixed Network Environment	Fraunhofer FOKUS, VELTI, Ericsson ETH
3	Auto-Configuration/Self-Configuration of Addresses in a SOHO Network	BME, Ericsson ETH
4	Auto-Configuration/Self-Configuration in MARSIAN	TARC-PL, Fraunhofer FOKUS
5	Auto-Configuration of radio channels in 802.11 networks	WUT, UL, TARC-PL
Autonomic Routing		
6	Autonomic multipath routing in 802.11 mesh networks	WUT, UL, TARC-PL
7	Autonomic Routing in Fixed Network Environments - Self-Adaptation of Routing as driven by Risk-Level Assessment in a Fixed Network Environment	ALBLF, BUPT, Fraunhofer FOKUS
Autonomic QoS, Mobility, Monitoring and Fault-Management		
8	Autonomic QoS Management in wired network	BUPT, ICCS, GRNET
9	Autonomic Mobility and QoS Management over an Integrated Heterogeneous Wireless Environment	ICCS, FLE, Ericsson ETH, VELTI, TID
10	Scenario for Network Monitoring and QoS Management	ICCS, TSSG, TID, TUB
11	Autonomic Fault-Management for selected types of Black Holes in a Fixed/Wired Network	FOKUS, TSSG, TARC-PL, ALBLF, Ericsson LMF, ICCS
12	Provision of Autonomic Services in self-configurable environments	GRNET, VELTI

In order to guarantee that WP5 goals are met a detailed scenario evaluation and assessment was performed and concluded to the following results.

Completeness

The defined twelve scenarios span over all the technical work packages and specifically:

- 7 scenarios come from WP2
- 2 scenarios come from WP3
- 3 scenarios come from WP4

The defined twelve scenarios span over the broad field of autonomic networking, as follows:

- 5 scenarios will demonstrate autonomic auto-configuration/auto-discovery
- 2 scenarios will demonstrate autonomic routing
- 5 scenarios will demonstrate autonomic QoS, mobility, monitoring and fault-management

Multimedia services and applications are deployed on top of EFIPSANS enabled autonomic networking environment both in the Standalone/Satellite and Integrated testbeds. These scenarios are:

- The Integrated (collaborative) scenario implemented in the integrated Testbed: Auto-Collaboration for Optimal Network Resource Utilization in Fixed Network Environment
- The Network Monitoring and QoS Management Scenario in the ICCS testbed
- The Self-configuration of Autonomic Services Scenario in the GRNET's standalone testbed

Heterogeneity

The twelve scenarios span over multiple networking environments:

- 7 scenarios will be demonstrated in a fixed network environment
- 4 scenarios will be demonstrated in a wireless network environment (IEEE 802.11)
- 1 scenario will be demonstrated in a hybrid heterogeneous cellular/WLAN network environment

The scenario template also includes the list of key players and their corresponding benefits. The final demonstrator will emphasize on this through the support of the visualizer and through appropriate presentation.

2.4 Testbed Integration Methodology

As highlighted in Section 2.3, the twelve scenarios defined by EFIPSANS span over heterogeneous networking environments, functionality and use-cases. In order to give a clear, uniform picture on the overall benefits of autonomic networking, the integration of some of these scenarios in a use case trial is required. The objectives of integration are the following:

- Harmonize the autonomic functions to be demonstrated with regard to inter-operability and networking environment
- Create a common testbed that can be used for experimentation
- Describe a high-level story-line for the scenarios

In order to fulfil the above objectives, first we selected scenarios that can be used in a given network environment. This is necessary since most autonomic behaviors in EFIPSANS are specific to a certain network environment, such as fixed, wireless or cellular. The grouping of scenarios ensures that each scenario is demonstrated in the appropriate environment.

Figure 2 shows how different scenarios were mapped to different networking environments. The yellow boxes represent the individual scenarios while in the bottom of the figure the magenta boxes indicate the networking environment.

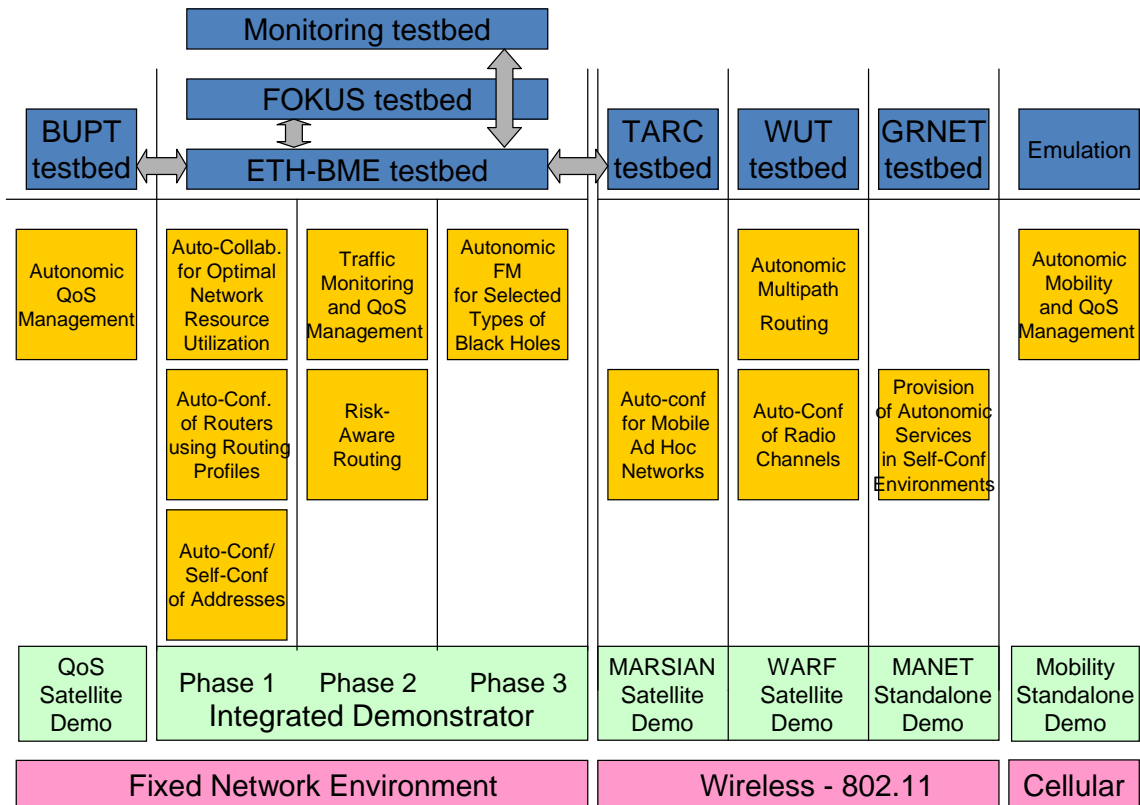


Figure 2: EFIPSANS Scenarios and Testbeds

2.4.1 The Integrated Testbed

One of the main objectives of EFIPSANS was to create a proof-of-concept testbed that can be used to demonstrate the autonomic functions researched and developed by the project. Since the consortium members are spread practically all over Europe, it would have required considerable effort to create an integrated testbed that is installed at a single geographical location.

However, the public Internet infrastructure enables a more or less straightforward interconnection of fixed networks. This motivated our decision to create a common integrated demonstrator core testbed that is composed of interconnected network segments (in Figure 2 depicted with the green box titled “Integrated Demonstrator”). This core testbed will host six out of the seven scenarios for fixed network environments.

The remaining scenario titled “Autonomic QoS Management in Wired Network” will be demonstrated in a satellite testbed (see Section 2.4.2) because of geographical and architectural reasons.

The core integrated testbed will be composed of the ETH-BME testbed located in Budapest, Hungary, the FOKUS testbed located in Berlin, Germany and the Monitoring testbed which aims at monitoring several nodes that are distributed among the partners. The interconnection of the networks will be based on a layer 2 tunneling solution, which enables passing both link layer and IPv6 packets (more details can be found in Section 3.1.1).

2.4.2 The Satellite Testbeds

In the design process of the wireless and cellular networking environments we face the same problem as before: it is not efficient to build a single wireless/cellular network that can be used to demonstrate all the functionalities defined in the project. Unfortunately, the interconnection of distant wireless and/or cellular network segments is not possible in a purely wireless manner (like we did in the core integrated testbed). Therefore we decided to demonstrate wireless-related EFIPSANS functionalities in isolated wireless network segments located at several partners. However, it is still advantageous to interconnect the wireless network segments by fixed links. For instance, several scenarios can make use of wireless network domains as access networks.

2.4.3 The Standalone Testbeds

Finally, we decided to demonstrate two EFIPSANS scenarios in a standalone manner. The “Autonomic Mobility and QoS Management over an Integrated Heterogeneous Wireless Environment” and the “Provision of Autonomic Services in Self-Configurable Environments” scenarios require highly customized network environment and network configuration, which would have been unfeasible to provide in an integrated way. These testbeds are independent from the integrated testbed.

2.5 Validation methodology

2.5.1 Concepts to validate

The EFIPSANS framework in principle aims to validate a number of fundamental (to autonomic engineering) features appropriately categorized for the purposes of this project per network type (fixed or mobile), functionality (layer-specific), topology (e.g. mobile ad hoc) and other (e.g. security). These features can be identified under the following five key concepts:

- Auto-Discovery
- Auto/Self-Configuration
- Autonomic Routing & Self-Adaptation
- Autonomic Mobility & QoS Management
- Autonomic Network Monitoring & Fault-Management

The validation of the former key concepts is required to assess their impact on the EFIPSANS framework and evaluate to what extent the defined R&D challenges and objectives coming from those key concepts were successfully addressed and implemented within the EFIPSANS project lifespan. In order to successfully complete such an assessment a common concept and step-based evaluation process must be specified to guarantee a smooth effective and unified evaluation.

EFIPSANS validation methodology incorporates a number of purpose-driven activities with specific expected outcomes such as:

1. Analysis of EFIPSANS specific documentation deemed suitable and essential in helping analyze the former key concepts

- Analysis of the EFIPSANS technical annex to identify if the identified R&D challenges at the project's start phase have been implemented into the underlying framework
- Identification and analysis of specific project deliverables which provide evaluations and recommendations on the adoption of key concepts in the individual work packages.
- Identification and analysis of project publications related to the key concepts to gain feedback and recommendations
- Analysis of the outcomes of the project related events (e.g. workshops) to get external insight on how the EFIPSANS key concepts were received and anticipated by the general public.

2. Completion and distribution of specific questionnaires

Completion of specific key concept questionnaires to be distributed to various groups in order to obtain feedback and recommendations of issues like:

- IPv6 vs. IPv4
- Autonomic systems awareness and usability
- Anticipation of autonomic behaviors and advanced functionality
- How the EFIPANS key autonomic concepts improve end user experience

- Benefits for Industry players (ISPs, Operators, SPs)
- Autonomic networking: Applicability, deployment and acceptance

3. Validation via Simulation

Validation of the specific key concepts via simulation is used as part of the overall evaluation of the EFIPSANS key concepts mainly to assess important issues around performance, stability and scalability that cannot be easily estimated in the project testbeds.

4. Analysis of the Qualitative/Quantitative tests and results

Selected Qualitative / Quantitative (Q&Q) tests and results directly related to the EFIPSANS scenarios that directly incorporate autonomic functionality related to the former key concepts, can be selected for analysis and evaluation. This can work towards identifying:

- What certain innovations have been validated in each scenario
- Recommendations / lessons we have learnt from the integration of DEs/DMEs that implement each key concept
- Identification of any problems encountered during pilot or productive system operation

5. Analysis of key concepts with respect to the EFIPSANS Business Model

The fact that existing EFIPSANS scenarios already analyze the business aspects around the corresponding key concept(s) they deal with (as part of the overall EFIPSANS business framework) creates itself a necessity to evaluate in practice the real impact of each autonomic concept in industry today. This will reveal the real value added by the EFIPSANS in the converged IP-based systems of today and tomorrow especially in the area of autonomic service management.

6. Analysis of the key concepts with respect to the standards

Circumstantial analysis surrounding standardization issues and any other impact the aforementioned key concepts have made in this area and particularly:

- Have the results or concepts coming from the project been taken up outside the EFIPSANS consortium?
- Which specific IETF drafts or standards have the key concepts contributed towards?
- How the adopted strategy achieved its purposes

2.5.2 Components to Validate

EFIPSANS demonstration framework aims to develop integrate and demonstrate the key concepts in a number of scenarios implemented in specific project testbeds. Figure 3 below illustrates how the developed components that realize the project key concepts are mapped within the different physical nodes per work package as software installed on these nodes in the testbeds. The grey box represents a physical node (e.g. mobile terminal, router, etc) and the colored boxes within each grey box represent WPx related components or packages that implement purpose-specific autonomic functionality (in the form of DEs/DMEs) on the top of existing functionality (e.g. QoS management).

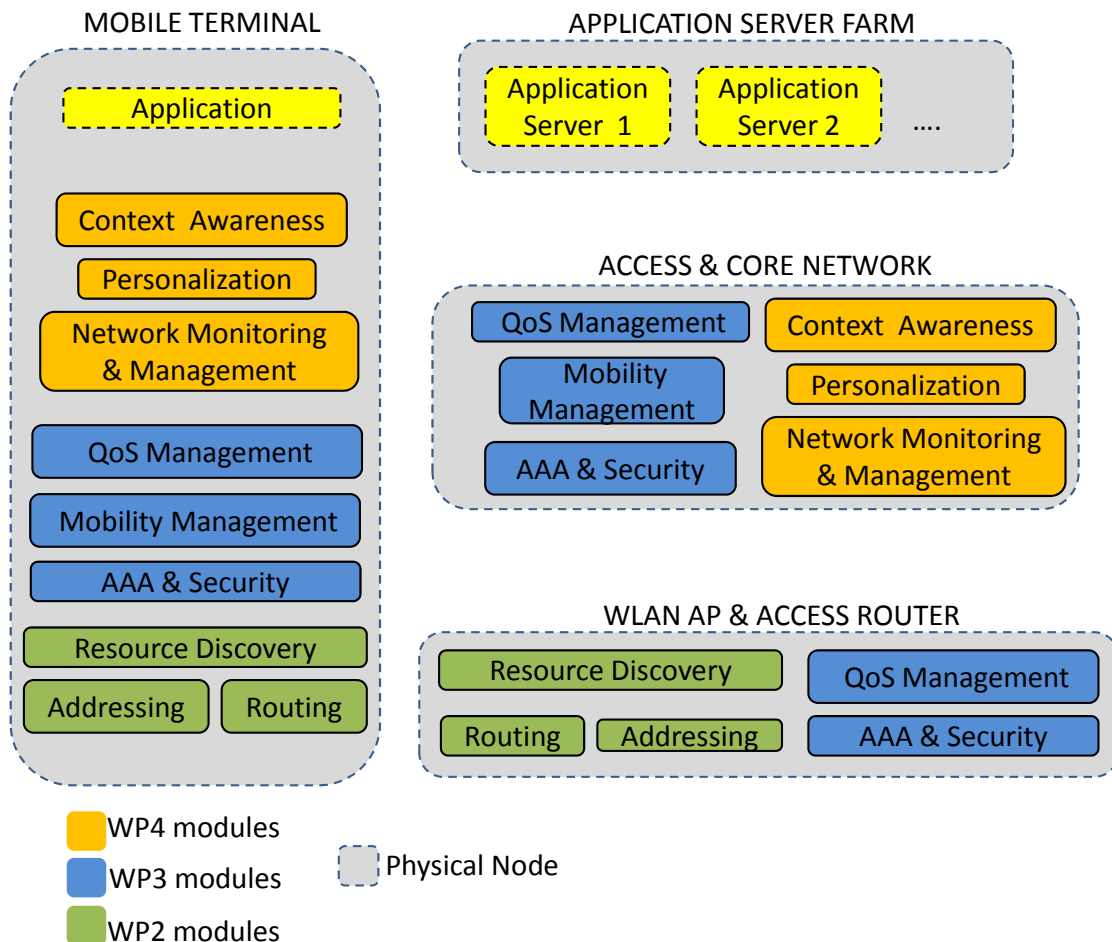


Figure 3: EFIPSANS Integration Environment

3 DESCRIPTION OF THE TESTBEDS

This Chapter gives an overview of the testbed components that will be shown in the final demonstrator. The description given here should be considered as preliminary, while more details will be specified in the next deliverables from WP5, as the implementation and testing work will be progressing.

3.1 Description of the Integrated Testbed

This section describes the integrated testbed architecture and the integration methodology for creating the integrated testbed.

3.1.1 Architecture

The integrated demonstrator is composed of interconnected testbeds at different partners. An overview diagram can be seen in Figure 4. The core of the integrated testbed is the ETH – BME testbed which is located in Budapest, Hungary. This testbed is connected to other partners' testbeds via tunnels.

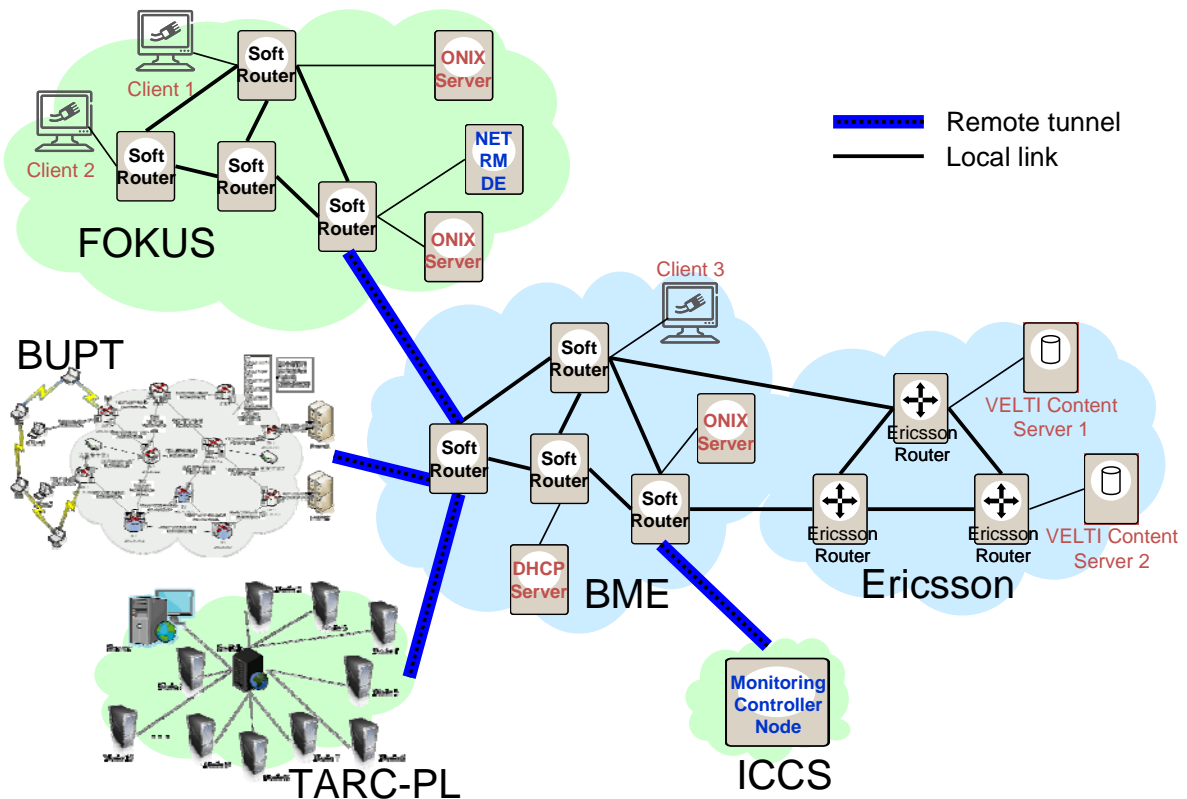


Figure 4: Integrated Demonstrator Architecture

The configured tunnels transfer layer 2 packets over IPv4 packets. The tunneling choice was motivated by the fact that the connectivity provided by the current Internet is still based on IPv4 dominantly. We chose to tunnel layer 2 packets so that in addition to IPv6 packets, link layer packets can also be exchanged between the tunnel endpoints. This provides a totally transparent connectivity on layer 2 and on layer 3, which is necessary to demonstrate some of the EFIPSANS scenarios (e.g., the DHCP smart relaying scenario). The chosen tunneling software is the open-source OpenVPN tool [5].

Two lightweight Streaming Content Servers implemented by VELTI will provide the application testing user interface environment that will demonstrate validate and test in real time how the Integrated autonomic networking system performs in defined scenarios when video streaming applications with different qualities are deployed. Demonstration of autonomicity in this case aims at proving how efficient the system reacts in certain problematic conditions faced by ISPs nowadays (e.g. congested links or paths, bandwidth instability, service redundancy, optimized resource management and utilization, etc).

3.1.2 Integration of the testbed components

In the following sections, the integration methodology used for creating the integrated testbed will be described.

3.1.2.1 Platform integration

The integrated testbed will primarily consist of routers, end-hosts and EFIPSANS-specific nodes (such as ONIX servers, network-level Decision Elements, etc.). It is an important integration task to harmonize the platform to be used for these nodes.

Basically, two types of IPv6 router nodes will be used in the testbeds:

- Soft-router nodes
- Commercial vendor routers

The soft router nodes will be based on the GNU Quagga [6] platform, which heavily builds on the GNU Zebra code base. In order to minimize the risks of inter-operability problems, we decided to use a common release (version 0.99.15).

Several commercial router nodes will be used in the integrated demonstrator. Three of them will be located in the ETH – BME testbed and will be based on the Ericsson SmartEdge 800 commercial router product [7]. These routers will be modified to include several EFIPSANS-defined protocol extensions and algorithms and will be connected to other partners' routers and nodes. An important integration task is to ensure the interoperability of these router nodes with the soft-routers, which we will address with functional tests [see Section 3.1.2.3].

3.1.2.2 IPv6 integration

Nodes in the integrated demonstrator will use site-scoped IPv6 addresses with several subnets defined. Except for a few nodes, the IPv6 address allocation and configuration will be handled by the Autonomic DHCP Architecture (ADA) defined by EFIPSANS.

The integrated demonstrator will use router nodes that are all capable of using the OSPFv3 routing protocol. The exact routing configuration will be automatically determined by the scenario titled “Auto Configuration for Routers using Routing Profiles”.

3.1.2.3 Functional integration and validation

The functional integration of the components from different scenario implementations must be based on the interface specifications of the components and the architectural constraints of the deployment of those components. The first one relates to the design time functional integration, the second one relates to the run time functional integration. While each scenario implementation will have its own set of test cases for the verification and validation of the proper operations, in course of integration of different scenarios a new set of test cases must be specified.

Most of the scenarios implemented focus on a single autonomic behaviour. The Integrated Demonstrator implements a bunch of autonomic behaviors and this gives the opportunity for testing the interactions among different autonomic behaviors. While there are no formal proofs of the consistent and efficient operations of the coexistent autonomic behaviors, there are means for testing the system before delivering it and/or some means to monitor the system states for detecting any malicious behaviour like oscillating control loops.

Examples of functional integration requirements are:

- Using the ADA architecture for bootstrapping the ONIX system
- Integration of the scenarios that manipulate routing

3.1.2.4 Integrated visualization

The goal of the visualization is to offer a centralized common view to better follow and understand the demonstrated mechanisms. This approach increases the efficiency of the dissemination activities, because the complex and distributed solutions provided by EFIPSANS are displayed on a single monitor easy to comprehend. The viewer of a demonstration can get an overall picture on the topology of the demonstration and on the structure of each node of the network. Additionally it makes possible to show and explain events occurred within a very short time period (i.e., orders of milliseconds), since lets the viewer to browse the logged event notifications.

The visualization is not intended to assist the implementation of the software modules or to help debugging the network configuration. Nevertheless, we expect that, in the last phase of integration, and especially in the case of reconfiguration/extension of the testbed. After a first phase of development and testbed integration, as the visualization framework is integrated with the demonstrated mechanisms it will be validated with basic operations, ensuring that the displayed information do indeed reflect the demonstrated mechanisms. Then after the setup of the testbed (i.e., connecting and powering on nodes, starting user-space applications), a centralized view on the inter-node communication and testbed events helps monitoring the chain of interactions and eases the task of identification potential points of failure.

The visualization framework consists of a centralized monitoring node and a notification function, which is implemented in each visualized Decision Element (DE) or Managed Element (ME). The visualization (monitoring) node runs the *visualization tool*, which is a Java application. The visualization node is linked over dedicated “monitoring” links to each visualized node. For remote nodes that are not located on the demonstration sites, this is replaced by TCP/IP connections over the public Internet, that is, we do not use/load the IPv6/IPv4 tunnels for visualization purposes. Each visualized DE or ME from the visualized nodes has to maintain its own TCP/IP connection to the visualization tool. Since we visualize only events, and each event notification will use only one IP packet to notify the visualization tool, the generated traffic will not congest the network.

Figure 5 presents a typical deployment of the visualization tool into the testbed. The central node running the visualization tool at the bottom of the figure is directly connected to the nodes (e.g. routers, servers, clients), thus the notifications reach the visualization tool without being congested and with minimal delay (less than 1 millisecond). This allows the prompt display of the events. The implementation is mainly based on sockets. Furthermore, the nodes from remote networks can also notify the visualization tool about their events over the Internet. In such case there might appear additional propagation delays, which depend on the specific location of the demonstration. The visualization tool is not replacing the monitoring or the management tools. It is just a means to make the demonstrated events more understandable. Therefore in a regular network the links from the nodes to the visualization node are not used, which means that these links do not interfere and do not affect the demonstrated events.

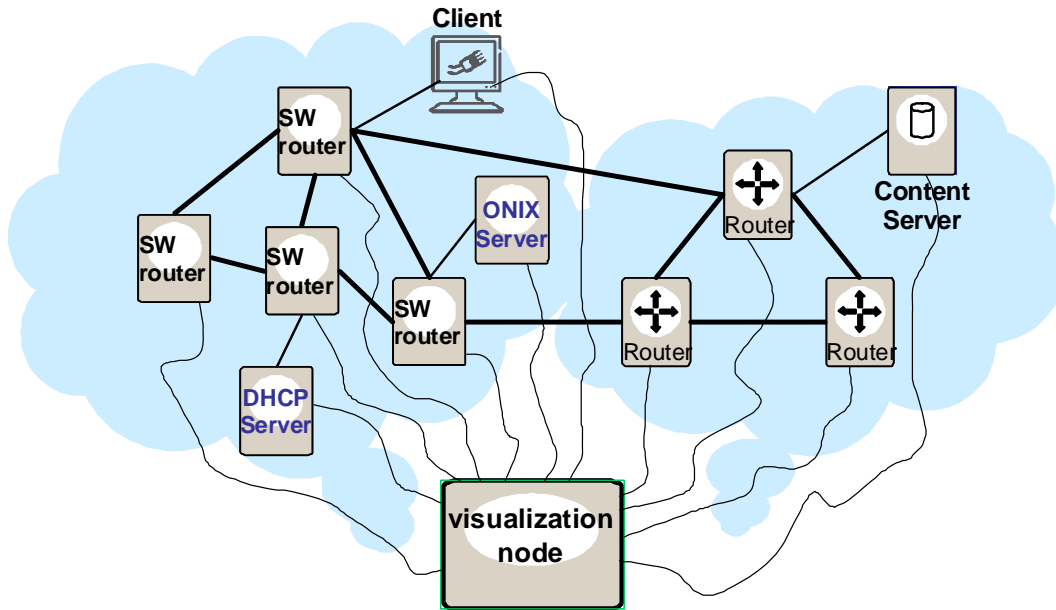


Figure 5: Visualization framework

Figure 6 presents a snapshot of the Graphical User Interface (GUI) of the visualization tool. The screen layout is divided into 2 main parts – one for visualize the nodes and one for the last few arrived event notifications. On the main screen the nodes are displayed in different colours/shapes depending on the type of the node (e.g. edge router, DHCP server, client, etc.). On the mouse over event it displays a few information about the node: IP address, ports, etc.. On clicking on the icon of the node a popup window appears with detailed information about the node (status, active DEs and MEs, routing protocols, etc.) In the list of events the last few events (3-5) are displayed in a readable format with the most vital information about it. However, all the events are logged, in order to aid the offline debugging the demonstrator. The Node List window is maintained for later use.

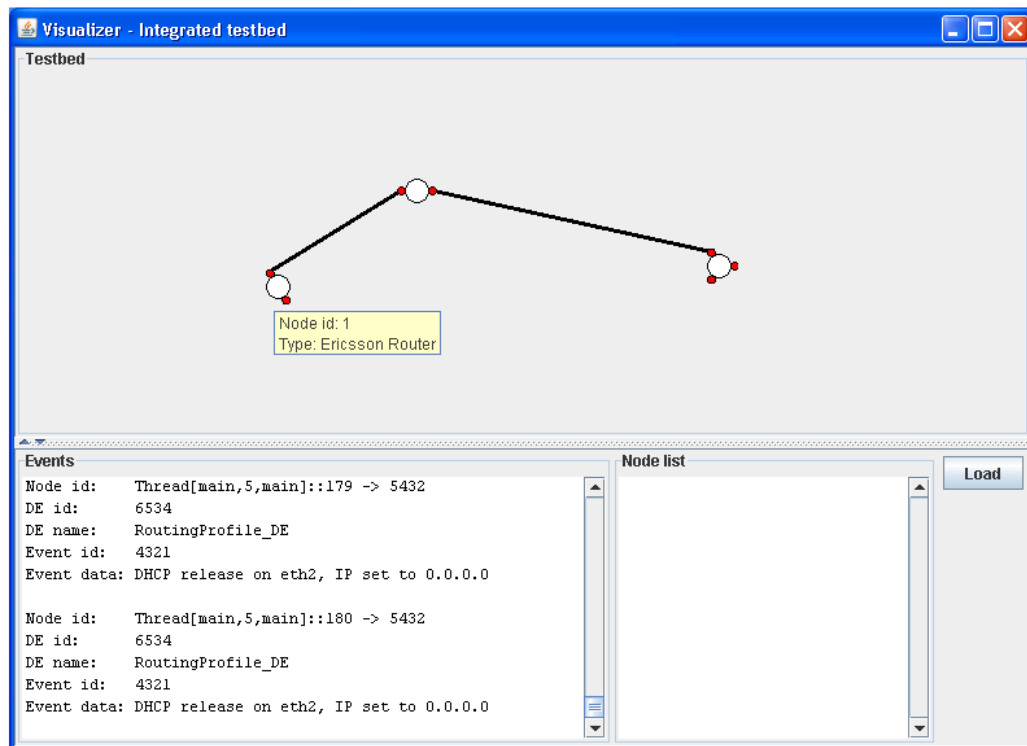


Figure 6: Visualizer tool – Graphical User Interface

The main canvas shows the physical topology, displaying the network interfaces belonging to different subnetworks in different colours. The DEs and MEs running on a node are displayed in a pop-up window if the viewer double clicks on a given node. The pop-up window displays only those entities that have established a connection with the visualization tool – thus only the active entities are visualized.

DE and ME IDs, notification format

Each DE should connect via a TCP socket to the visualization. Then it will have to send a char string each time an event occurs. The coding is UTF-8.

The event notification has 5 fields.

- *NodeID* - The node's unique identifier of the node where the event occurs (not the IP address because a node can have more interfaces therefore more than 1 IP addresses).

With this ID the Visualizer Tool knows which node the event occurs on (will be defined as we agree on the demo network topology)

- *DE_name* - DE name to be displayed. Describes the type of the DE which helps us to know which DE should be assigned to the node. This information is redundant, because from the previous 2 IDs we can learn this information.
- *Event_ID* – it is an ID that uniquely identifies the event type (e.g. the "DHCP release" will be 4321 on each node, for each DE)
- *EventData* - a text to be displayed in the event window

The exact format is the following:

NodeID (nums, 4 digits)|DE_ID (nums, 4 digits)|DE_name (chrs, max 25 chrs)|Event_ID (nums, 4 digits)|EventData (chrs, 250 chrs)\r\n

Example: 5432|6534|DHCPv6_NODE_DE|4321|DHCP release on eth2, IP set to 0.0.0.0\r\n

3.1.2.5 Illustrations of other Visualization Methods to be considered for GANA

EFIPSANS will also consider different visualization methods that can be employed in visualizing some advanced interaction flows within the GANA framework. For example, interaction sequences between DEs at different levels within GANA, as well as their autonomic behaviors executed on the “management interfaces” of their assigned Managed Entities (MEs). However, it is worth noting that it is a hard task to develop an advanced visualizer, and so attempts will be made to go for the minimum that is achievable. In this section, we provide a few examples of how GANA can be visualized in different contexts of its instantiation.

Example 1: *Visualization of GANA in the context of QoS and Mobility Management in a heterogeneous environment*

As discussed in the previous section, a graphical interface demonstrating the detailed operation of a node in terms of DEs is required. Here, we describe how we envision the graphical representation of the emulator for the integrated QoS and Mobility Management Scenario, showing the items that should be visualized in the particular scenario.

LOAD:

WLAN_1: 5.32
WCDMA_1: 2.87
WCDMA_2: 6.39

ATTACHED USERS:

WLAN_1: 4
WCDMA_1: 12
WCDMA_2: 9

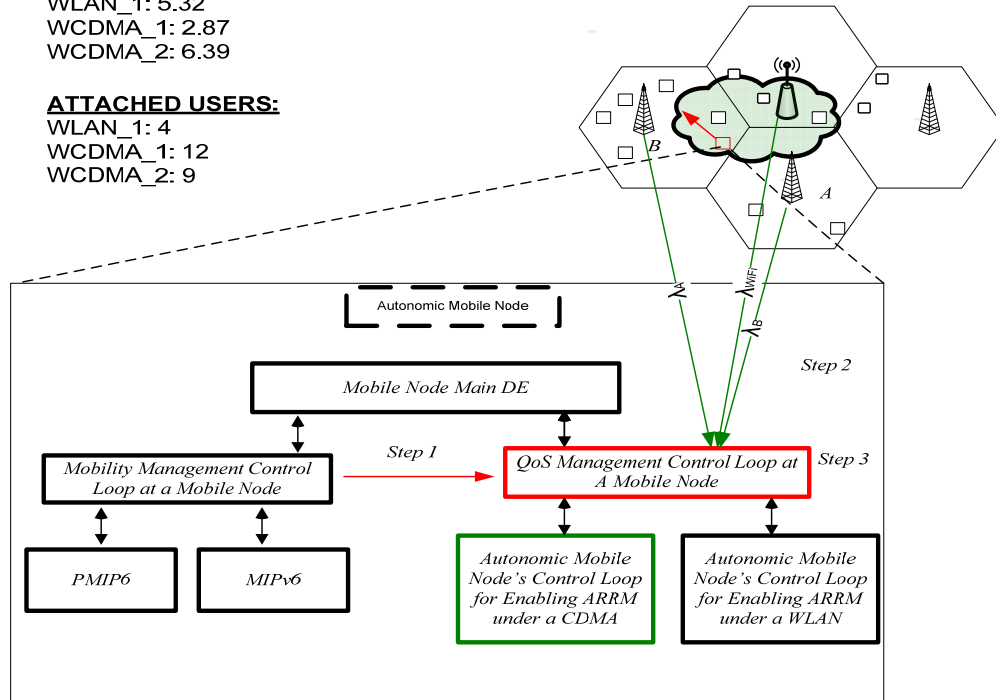


Figure 7: Visualization of DE interactions involved in the autonomic QoS and Mobility Management Scenario

The graphical interface should be able to present the following items:

- Networks topology,
- Network metrics such as traffic, load, attached users etc.
- The DEs that participate any time instant in any autonomic self-* mechanisms and their corresponding interactions. Specifically, a graphical interface is needed providing a DE view off all nodes, giving the possibility to depict step by step how an event progresses through illustrating the cooperation and realization of all control loops, of all nodes participating at the specific event.

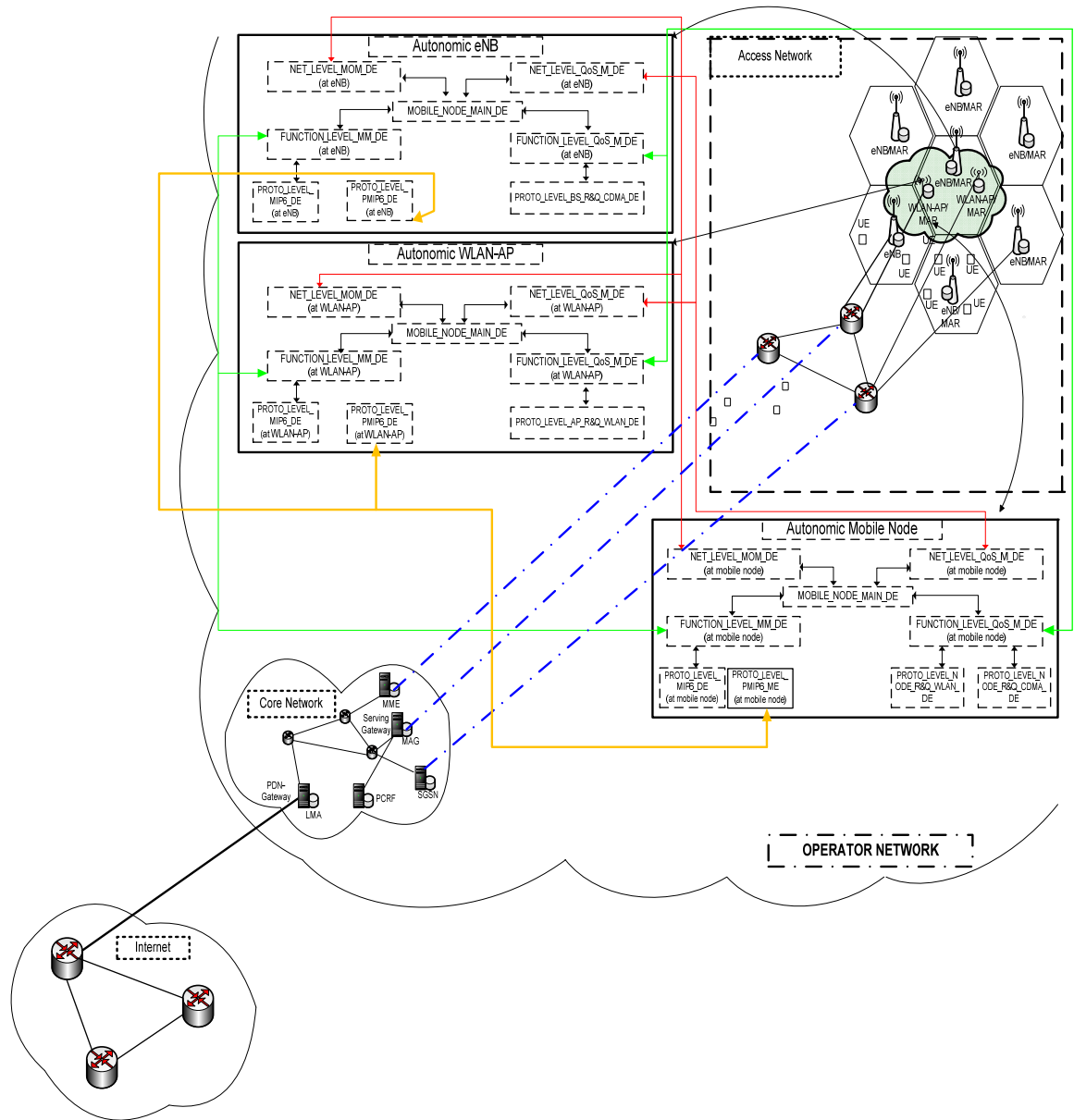


Figure 8: Advanced visualization of DE interactions in the autonomic QoS and Mobility Management Scenario

Example 2: Illustration of visualization of synchronization of actions in GANA in the context of autonomic Fault-Management and Resilience and Survivability within a node

Figure 9 below, we illustrate how a sequence of DE interactions can be visualized to the point of a Function-level DE issuing a command to a Managed Entity (ME) at the lowest layer/level of GANA.

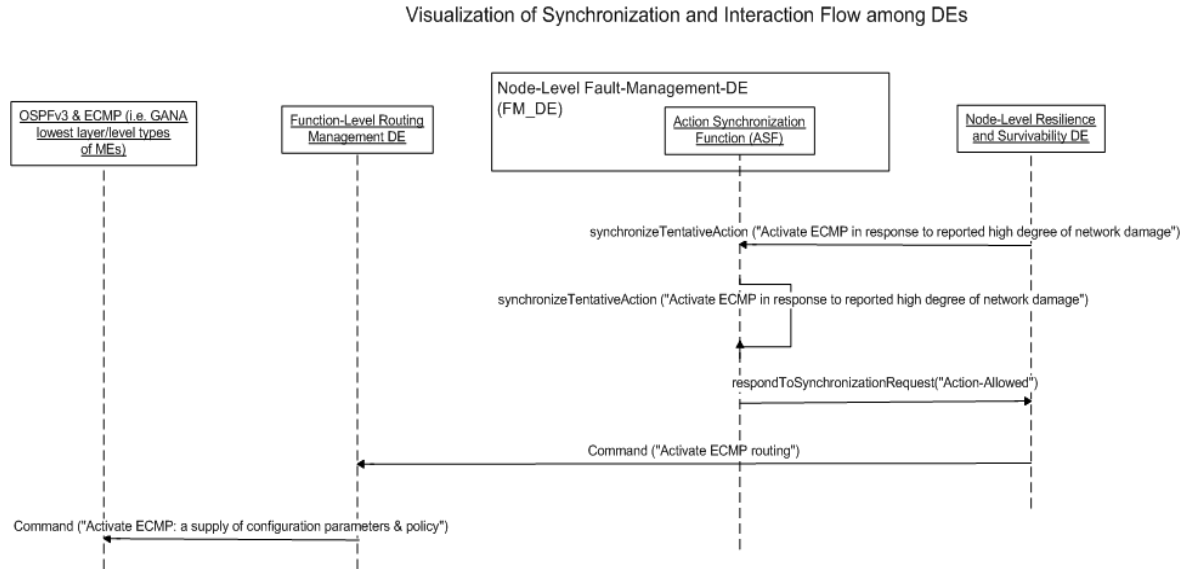


Figure 9: Visualization of synchronization of actions in autonomic Fault-Management and Resilience and Survivability within a node

3.2 Description of the Satellite Testbeds

3.2.1 BUPT testbed

Goal

For the demonstration purposes of selected aspects of the scenario on “Autonomic QoS Management in Wired Network” BUPT has planned a testbed platform. Through the proposed scenario the aim is to demonstrate the following self-* functionalities:

- Self – adaptation of the appropriate mechanisms such as choosing the appropriate packet marking mechanism etc. in terms of the dramatically changing network environment.
- Self – configuration of the parameters of the mechanisms, such as the maximum packet loss rate in the queue management mechanism etc, according to the current network status, node status, and the upper level ABs.
- Self – optimization of the network performance in terms of the view of both the node and the network. For instance, when the traffic load of a flow of a certain priority is always very high, the optimization-decision maker, NET_LEVEL_QoS_M_DE may decide to adjust the bandwidth allocation scheme of entire network.

Testbed topology and structure

The demonstration will be based on the following network topology, as shown in Figure 10

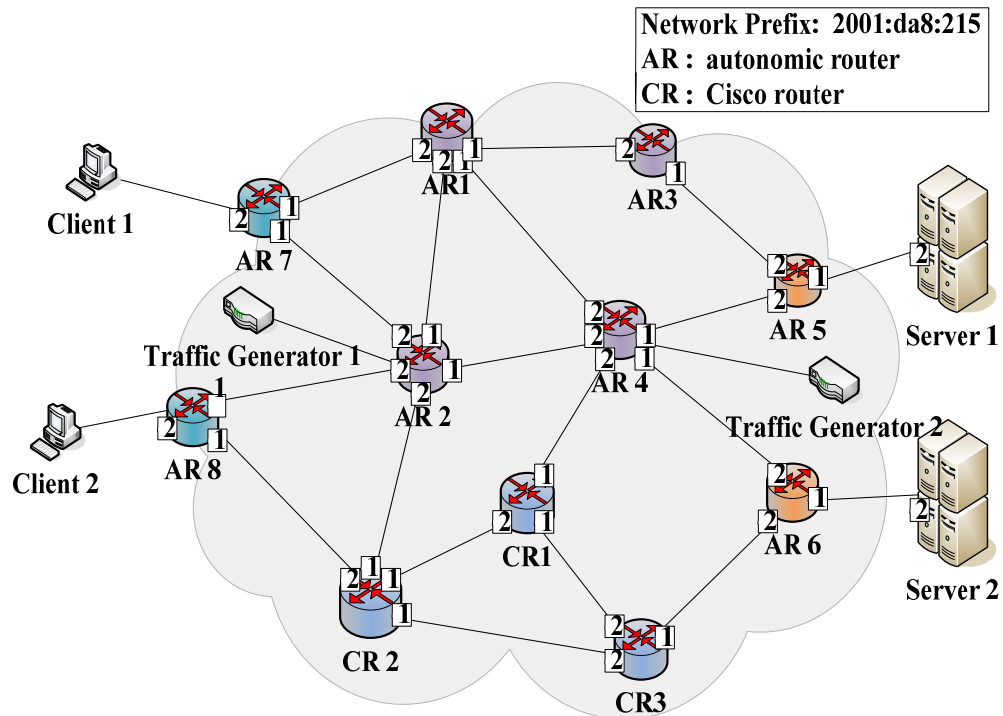


Figure 10: BUPT testbed topology

In Figure 10, 8 ARs (Autonomic Routers) and 3 CRs (Cisco Routers) are used. ARs implement the functions of Autonomic QoS management in the GANA architecture. The AR5-AR8 are the edge nodes, they implement NET_LEVEL_QoS_M_DE, FUNC_LEVEL_QoS_M_DE, PROTO_LEVEL_SA_DE, PROTO_LEVEL_PM_DE, PROTO_LEVEL_QM_DE and PROTO_LEVEL_QS_DE, while the AR1-AR4 are core nodes, they implement two DEs, i.e., PROTO_LEVEL_SA_DE and PROTO_LEVEL_PM_DE.

CRs do not have the autonomic mechanisms, but they can be used to test and verify that the implemented autonomic systems are compatible with both the autonomic and non-autonomic routers.

In addition, we use traffic generator to generate various levels of flows in order to test the performance of our program.

3.2.2 MANET testbed

Goal

For the purposes of demonstration of selected aspects of the scenario on “*Auto-configuration for mobile ad hoc networks*” Telcordia (TARC-PL) has created a testbed platform. It is intended to enable emulation and/or simulation of real-world wireless mobile ad hoc network (MANET) environments. The main feature demonstrated on TARC-PL testbed will be the functionality of EFIPSANS-proposed IPv6 extension: ND++ protocol [3].

Testbed topology and structure

TARC-PL testbed (see Figure 11) contains 15 nodes in the form of diskless workstations and one server. Physically all stations within a testbed are connected to the server by means of switching device. Each node has two network interfaces – wired Ethernet used for testbed maintenance and configuration – and wireless card used to create wireless network within the nodes.

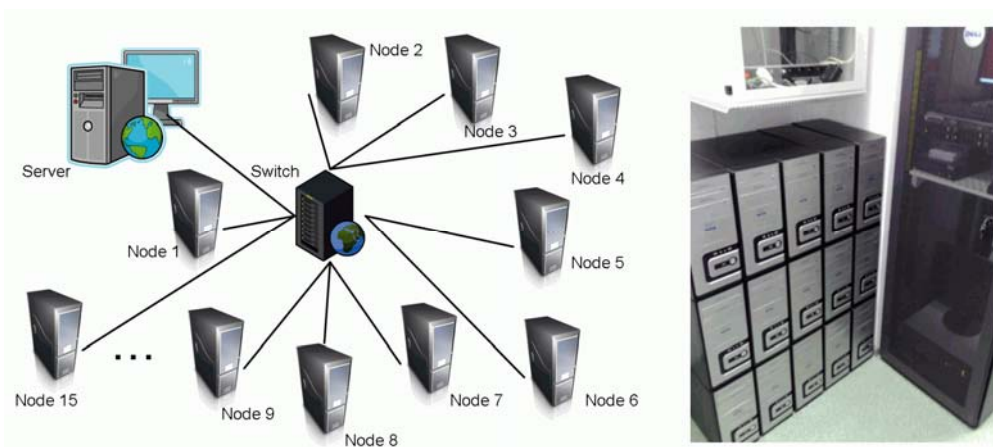


Figure 11: TARC-PL testbed: physical network topology from wired Ethernet interface perspective and testbed setup overview

Testbed nodes are booted as diskless workstations:

On system start-up wired ethernet cards localized at each node obtain an IP address and localization of boot loader file from DHCP server, which consist necessary information for boot process start up. In initial part of testbed setup IPv4 address is used, it is though clear that final emulations will be held in IPv6 environment. It would be possible to use IPv6 in both cases, however due to modifications in IPv6 kernel code we prefer not to do this in order to separate testbed setup from demonstration results. Based on information gathered at each remote node, FreeBSD system is booted and a mounting point to the file system localized at the server is created. File system is built on the basis of operating system's kernel which can be reloaded after introducing specified changes.

Logical structure of the testbed (see Figure 12) comprises the emulated wireless mobile ad hoc networking environment, similar to one presented in Figure 12. In consequence the testbed is able to emulate multi-hop real-world MANET scenarios. In order to emulate multihop visibility and mobility of nodes two methods were investigated: power control of wireless interfaces aimed at emulating nodes approaching, receding, loosing connection, etc., and IP packet filtering. Initial test results demonstrated that first method is not suitable for this kind of setup due to very close antenna placement. Thus the second method was implemented. IPFW packet filtering rules are applied to set up firewall IP packet blocking at each node, specified according to envisioned network situation.

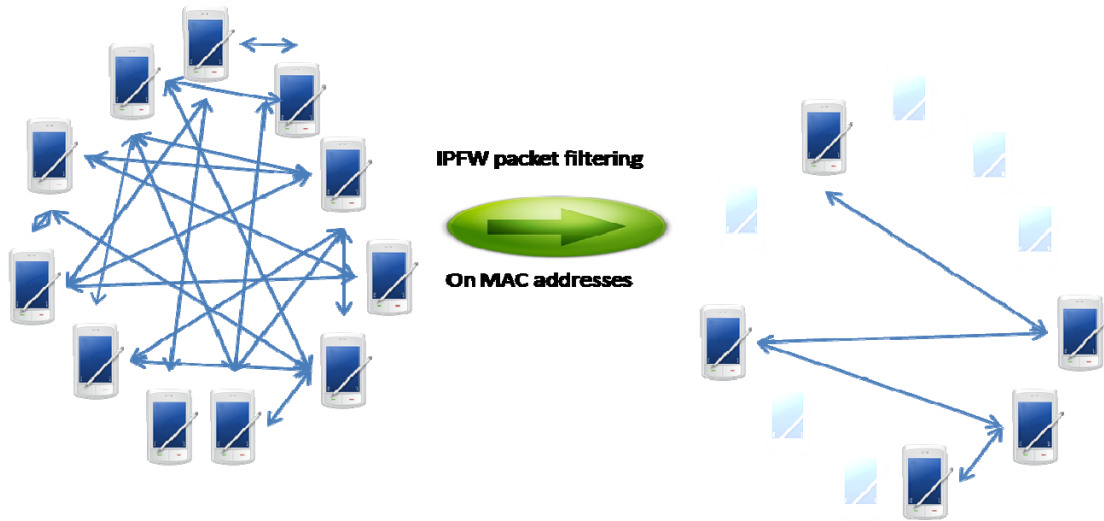


Figure 12: TARC-PL testbed: physical network topology from wired Ethernet interface perspective and the resulting multi-hop MANET

Hardware and software specification

Hardware:

- Each of 15 nodes is a diskless workstation with the parameters given below:
 - Processor Intel Core2 Duo E7200 2,53GHz (S775)
 - DRAM memory: Kingston DDR2 2GB 800MHz CL5
 - 802.11g wireless network card: D-Link Wireless PCI DWL-G510
- Server taking over also the role of DHCP server contains 2 HDD drives:
 - 2 x Seagate 250 GB Barracuda ES.2 (32MB, Serial ATA II)
- KVM switch for console access

Software:

- FreeBSD operating system (Release 7.0) with KAME-like IPv6 implementation enabling access to the source code

Capabilities

Telcordia's testbed is based on the FreeBSD system which enables access to the source code of a system with implemented IPv6 features. It is possible then to apply some extra functionality to existing protocol implementations or to enhance the system kernel with some new features specified by needs of experiments. Currently the implementation of EFIPSANS proposed Neighbor Discovery extensions is ongoing and first demonstrations were performed presenting currently implemented functionality.

3.3 Description of the Stand-alone Testbeds

3.3.1 Autonomic Mobility and QoS Management

EFIPSANS scenario "Autonomic Mobility and QoS Management over an Integrated Heterogeneous Wireless Environment" aims at demonstrating some basic self-* functionalities of an autonomic mobile node which are enabled by the proposed corresponding architecture, as well as to reveal the emerging benefits from both users' and networks' perspective. To facilitate our goal, the demonstration in a testbed/emulator key of autonomic functionalities concerning mobility management and resource allocation over an integrated cellular/WLAN network is envisioned. The complexity of the proposed testbed, due to the inherent characteristics of the wireless environment, led us to the creation and exploitation of a stand-alone testbed/emulator, which main actors will be Wideband-CDMA (WCDMA) cells, WLAN (WiFi) and autonomic mobile nodes. The key functionalities that will be implemented and demonstrated are presented in the corresponding "Scenario Description Template" at the end of this deliverable. In the following we place emphasis on summarizing some key technical aspect of the proposed testbed.

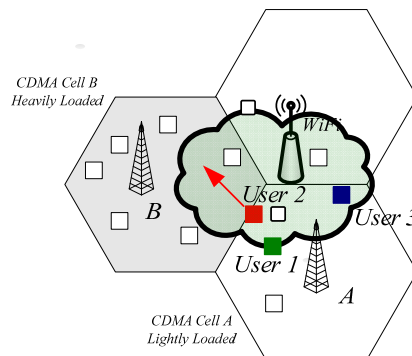


Figure 13: The scenario's environment and key actors

Figure 13 illustrates the key players in the proposed scenario, which include: a) two WCDMA base stations and their corresponding cells namely, cell A which is assumed to be lightly loaded and cell B which is assumed to be heavily loaded, b) a WiFi access point and its corresponding access area, and c) three autonomic users (i.e User 1, requesting data, non-real-time services and User 2 and User 3 requesting real-time multimedia services). Our goal is it to create such network and users conditions that will trigger some specific self-* functionalities of the autonomic mobile nodes under consideration.

To facilitate our goal, in Figure 14 we illustrate a simplified map of a 3GPP architecture and hence, we indicate (with red squares) the network components of the architecture that will be emulated via the proposed testbed (i.e. only part of their actual functionalities will be implemented).

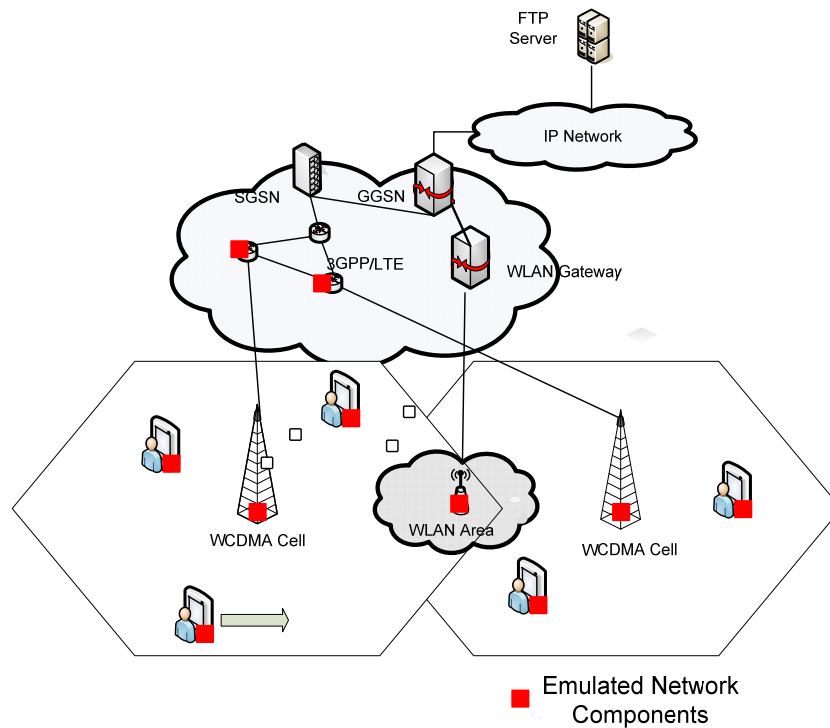


Figure 14: Mapping our Wireless Emulator to a Real Architecture (3GPP/LTE)

To that end, in the following we summarize some key technical details of the envisioned testbed, towards realizing the emulated environment illustrated in Figure 15

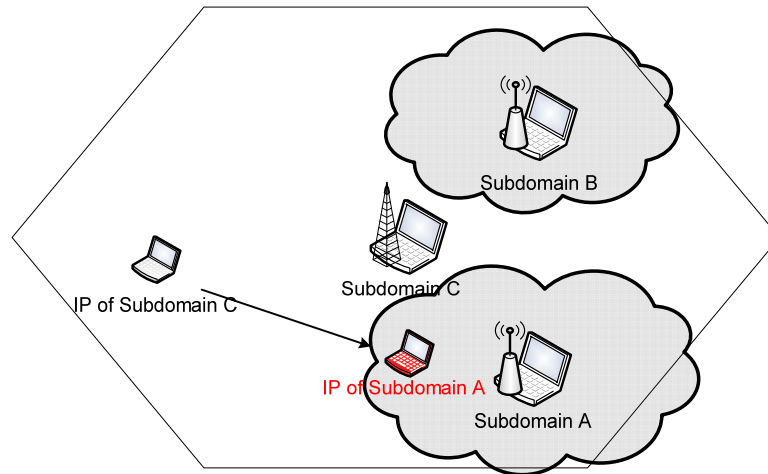


Figure 15: Emulated Wireless Environment

Hardware Requirements

- 5-6 PCs (notebook/desktop) with MATLAB [16]

Communication

- All PCs connect through a 802.11 AP/router and acquire a IPv4 or IPv6 address

Scenario realization

- Every PC has a specific role (i.e. W-CDMA Base Station, 802,11 access point, mobile node). Each AP or BS belongs to a different subdomain.
- Each node should be able to communicate using the 802.11 network with all emulated AP or BS in its locality and thus enable its autonomic functionalities.
- Partially implemented/emulated heterogeneous wireless network consisting of WCDMA base stations, WLAN access points and mobile nodes with multimode capabilities (all in different computers).
- All scheduling and resource allocation algorithms run at local MATLAB [16] enabled algorithms and all utilization and performance metrics come from the environment.
- All “Network Selection” algorithms run at local MATLAB [16] enabled algorithms and the attachment decision is disseminated to the real network for realization.
- Input required for the MATLAB [16] enabled algorithms (position of the mobile nodes, network specific constants as mentioned in section B of this document) will be provided by the real network.

Graphical Interface

A graphical interface demonstrating the detailed operation of a node in terms of DEs is required. In the following we provide a conceptual description on how we envision the graphical representation of the previous described emulator.

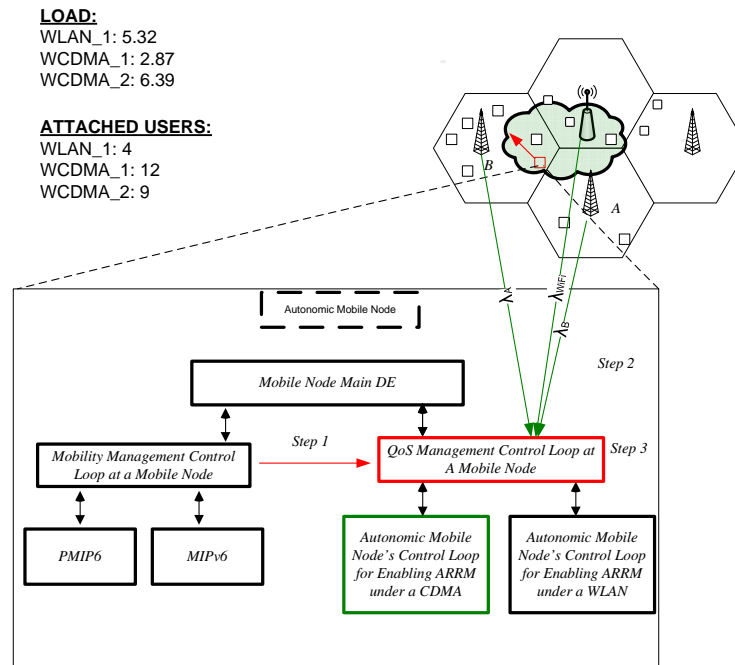


Figure 16: Example GUI for the scenario

Figure 16 shows an example graphical interface representing:

- Networks topology
- Network metrics such as traffic, load, attached users etc.
- The DEs that participate any time instant in any autonomic self-* mechanisms and their corresponding interactions. Specifically, a graphical interface will provide a DEs view of all nodes, giving the possibility to depict step by step how an event progresses through illustrating the cooperation and realization of all control loops of all nodes participating at the specific event.

3.3.2 WARF testbed

EFIPSANS WARF scenarios “Auto-Configuration of radio channels in 802.11 networks” and “Autonomic multipath routing in 802.11 mesh networks” (can be found in the Appendix) will be validated by simulations and small scale implementations.

For simulations OMNET++ and ns-3 will be used. For small scale implementation several single radio channel nodes based on WRT-54 Linksys hardware and Mikrotik multi-channel embedded board will be used. The number of used nodes because of practical reasons (power supply, the density and physical network size) will be limited to approximately 10.

3.3.3 GRNET testbed

The EFIPSANS scenario “Provision of Autonomic Services in self-configurable environments” will be validated by simulations and small scale implementations.

This scenario refers to autonomic network visualization, performance monitoring and resource management in fixed and wireless networks. In wireless environments, mobile ad-hoc networks (MANETs) are being created from neighbouring nodes and communication is established through IPv6 connectivity. In fixed environments, different LANs may be connected or disconnected and the provided monitoring services are updated in an autonomic manner. In each case, the network administrator desires to have a continuous view of the network topology and also information about the current network status (established links, available paths, and performance metrics) and the available resources.

The scenario may be demonstrated through the use of several laptops, mobile phones and PDAs. All of them will be automatically connected to the overlay network and each one of them will be able to provide the designed services (Figure 17). No special requirements are imposed to the testbed.

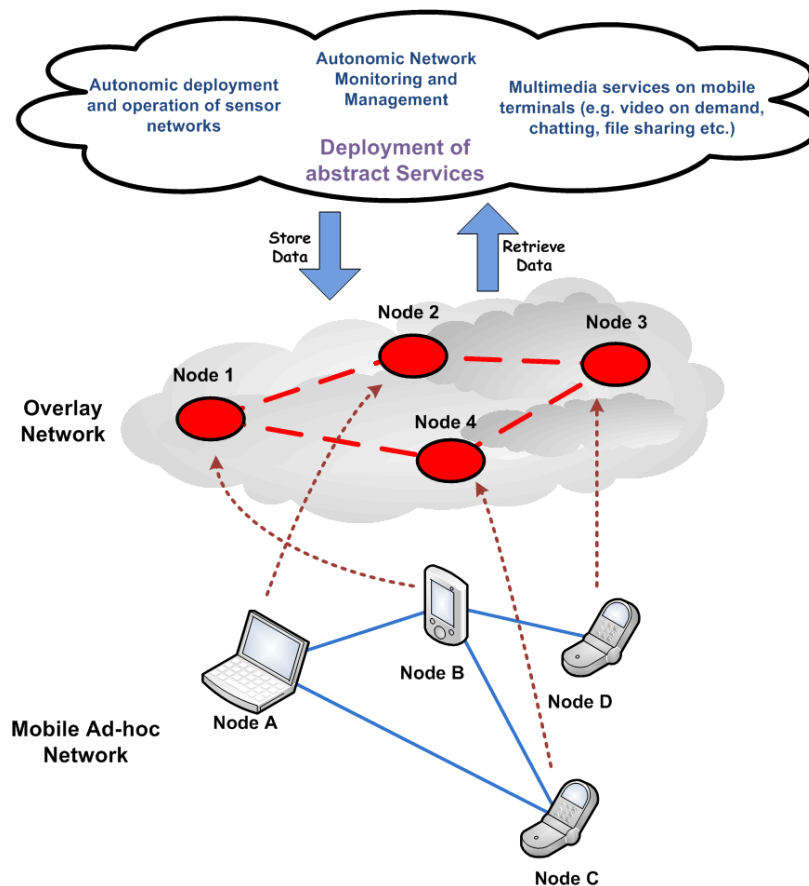


Figure 17: Autonomic Provision of Services

4 SPECIFICATION OF THE FINAL DEMONSTRATION

4.1 List of demonstrated protocol extensions

The following main IPv6 protocol extensions (see [3]) will be demonstrated in the frame of the Integrated Demonstrator:

- DHCPv6++
- ICMPv6++
- ND++
- IPv6 QoS Header Extension

4.2 List of demonstrated Components

The actual specification of the Decision Elements can be found in [2]. The final version of the document will be available at the end of the project.

Table 2: List of NETWORK LEVEL DEs

NET_LEVEL_AC_DE	Network-Level Auto-configuration Decision-Element - this AC_DE must oversee and control the overall situation in the network with respect to auto-configuration and self-configuration
NET_LEVEL_RM_DE	Network-Level Routing-Management Decision-Element - This is a network level instance of the FUNC_LEVEL_RM_DE. The purpose/goal of this Network-Level DE is the same as the Function-Level RM DE abstracted to a network level with network wide information/view.
NET_LEVEL_QoS_DE	Network-Level Quality of Service Decision Element – the version used in the Core Demonstrator
NET_LEVEL_QoS_M_DE	Network-Level Quality of Service Decision Element – the version used in the Autonomic QoS Management in wired network satellite scenario.
NET_LEVEL_R&S_DE	Network-Level Resilience & Survivability Decision-Element - this DE is responsible for managing the resilient behavior of a node as a whole and provides generic solutions ensuring its resilience and survivability.
NET_LEVEL_FM_DE	Network-Level Fault-Management Decision-Element – see at the NODE_LEVEL_FM_DE

Table 3: List of NODE LEVEL DEs

NODE_LEVEL_QoS_DE	Node Level QoS Decision Element
NODE_LEVEL_MAIN_DE	Node Main Decision-Element - The purpose of this DE is to facilitate the Auto-Discovery functionality for the node. Further, it houses the other node-level sub-DEs, and provides them with a common interface to manage the function-level DEs. Moreover, communication of its sub-DEs with other nodes in the network is carried out through the Node-Main-DE. The NODE_MAIN_DE also facilitates the interaction between the node and ONIX.
NODE_LEVEL_AC_DE	Node-Level Auto-configuration Decision-Element - this AC_DE must oversee and control the overall situation in the node with respect to auto-configuration and self-configuration.
NODE_LEVEL_SM_DE	Node-Level Security-Management Decision-Element - detects suspicious and/or malicious activity and autonomously react to it.
NODE_LEVEL_R&S_DE	Node-Level Resilience & Survivability Decision-Element (as a part of the Node Main Decision-Element of an autonomic node/device) - this DE is responsible for managing the resilient behaviour of a node as a whole and provides generic solutions ensuring its resilience and survivability.
NODE_LEVEL_FM_DE	Node-Level Fault-Management Decision-Element (as a part of the Node Main Decision-Element of an autonomic node/device) - the main objective of this DE is to analyze and react upon fault/error/failure/alarm notifications related to all MEs and DEs in a node and the network as a whole, thereby realizing the process of automated Fault-Diagnosis/Localization/Isolation and eventually automated Fault-Removal.

Table 4: List of FUNCTION LEVEL DEs

FUNC_LEVEL_RM_DE	Function-Level Routing-Management Decision-Element - auto-management of routing protocols in a wireless nomadic node and in Fixed/Wired Networks using OSPF.
FUNC_LEVEL_MON_DE	Function Level Monitoring Decision Element - this DE must oversee and control the overall monitoring and management of monitoring MEs within a node. The DE will also coordinate between other FUNC_LEVEL_MON_DEs of peering nodes to coordinate monitoring tasks.
FUNC_LEVEL_QoS_M_DE	Function-Level Quality-Of-Service Management Decision-Element

	– this is the version used in the Autonomic QoS Management in wired network scenario.
--	---

Table 5: List of the PROTOCOL LEVEL DEs

PROTO_LEVEL_OSPF_DE	Protocol-level OSPF DE
PROTO_LEVEL_PM_DE	Protocol-level Packet-Marking Decision-Element – this DE enables the nodes self-adaptation mechanism in the wired core network in terms of configuring the parameters of the QoS protocols used to implement the packet mark function dynamically, according to the current network status as well as the ABs accepted from the up level DE
PROTO_LEVEL_SA_DE	Protocol-level Service Aware Decision-Element - SA_DE is the decision-making element of Service-aware management function, which can improve the accuracy and effectiveness of service identification in IP network.
PROTO_LEVEL_QM_DE	Protocol-level Queue-Management Decision-Element – this DE enables the nodes self-configuration and self-adaptation mechanism in the wired core network in terms of configuring the parameter of the QoS protocols used to implement the queue management function dynamically, according to the current network status (e.g. whether the network is congestion) as well as the ABs accepted from the up level DE.
PROTO_LEVEL_QS_DE	Protocol-level Queue-Scheduler Decision-Element – this DE enables the nodes self-adaptation mechanism in the wired core network in terms of configuring the parameter of the QoS protocols used to implement the queue scheduler function dynamically, according to the current node status (e.g. the occupancy rate of egress link in the node) as well as the ABs accepted from the up level DE.
PROTO_LEVEL_ND++_DE	Protocol-Level Neighbour-Discovery++ Decision-Element - Manage extended Neighbor Discovery Protocol (ND++) which is specified in [3].

Table 6: List of the Managed Entities (MEs)

OSPF_ME	OSPF ME
DHCPv6_SERVER_ME	DHCPv6 Server ME used in the Autonomic DHCP Architecture (ADA)
DHCPv6_CLIENT_ME	DHCPv6 Client ME used in ADA
DHCPv6_RELAY_ME	DHCPv6 Relay ME used in ADA
ND++_ME	Neighbor Discovery++ ME used in e.g., Auto-configuration/Self-configuration in MARSIAN
NIC_ME	Network Interface Card ME used in almost all scenarios
STREAMING_APP_ME	Darwin Streaming Server ME (http://dss.macosforge.org/)
BART_ME	Bandwidth Availability in Real-time ME
EB_ME	Effective Bandwidth ME
AC_ME	Admission Control ME
QoS_V_ME	QoS Violation ME
PC_ME	Packet Classifier ME
PM_ME	Packet Marking ME
SA_ME	Service Aware ME
QM_ME	Queue Management ME
QS_ME	Queue Scheduler ME

4.3 Overview of the final demo

In this section, a high level overview of the individual steps in the final demonstration will be given. In the current stage of the work, the below steps should be considered as a preliminary example only, while a more precise description of the technical details will be presented in the next deliverables of WP5.

Core integrated testbed

1. The demonstrator network is plugged together, nodes have been turned on and are booting up
2. The scenario titled “Auto-Configuration/Self-Configuration of Addresses in a SOHO Network” is triggered and takes care of automatically configuring site-scoped IPv6 addresses in the network
3. Nodes discover the ONIX system
4. The routers publish their capability description to ONIX and receive a routing profile according to the scenario titled “Auto-Discovery and Auto-Configuration (Self-Configuration) of Routers using Routing Profiles in Fixed Network Environment”
5. A video streaming application example is shown using the “Auto-Collaboration for Optimal Network Resource Utilization in Fixed Network Environment” scenario, where the streaming content server is chosen automatically depending on the actual load of the network and the server load.
6. An overheating event is emulated in one of the routers. The “Autonomic Routing in Fixed Network Environments - Self-Adaptation of Routing as driven by Risk-Level Assessment in a Fixed Network Environment” scenario will be triggered and re-routes the traffic to minimize the risks of connectivity problems (e.g., packet loss).
7. A black hole event is generated and triggers the scenario titled “Autonomic Fault-Management for selected types of Black Holes in a Fixed/Wired Network”. It automatically detects the problem and resolves it.
8. Monitoring components are continuously providing information for the different scenarios during the demo. In addition, a special call admission control scenario utilizing the EFIPSANS monitoring framework is demonstrated.

During all the above steps, the Visualization Tool described in Section 3.1.2.4 will provide a continuous feedback about the events occurred and about the actual state of the demonstrator network.

Satellite testbeds

The satellite testbeds will be contributing to some of the steps demonstrated in the core integrated testbed, mostly acting as an access network. In addition, they will demonstrate their own scenarios in a independent manner, as explained above.

Standalone testbeds

Scenarios in the standalone testbeds will be demonstrated in an independent manner, as highlighted in previous subsections.

5 CONCLUSION AND NEXT STEPS

This deliverable describes the scenario development and selection process for the Integrated Demonstrator. It contains the description of the core integrated testbed, the satellite testbeds and the standalone testbeds in a comprehensive manner. A chapter is also given on the validation methodology. That will be followed throughout the final implementation and demonstration phase. The description of the structural components and a draft story-line for the Integrated Demonstrator is given as well. Finally, the Appendix contains the scenario descriptions.

There is an intensive work on the prototype development involving several partners at the various WPs. The WP5 organizes several Integration Workshops to put together those developments into either an Integrated Demonstrator or a common framework. The development and integration process is split into phases: the Phase1 is planned for end of June. The satellite and stand-alone scenario developments are running in parallel as well. The Phase2 of the Core Demonstrator development together with the integration with the two satellite demonstrators will be completed in September 2010. The same month the deliverable D5.2 will be published. The rest of the integration work and deliverable D5.3 will be concluded in December 2010.

REFERENCES

- [1] EFIPSANS project Deliverable D1.5 “Third Draft of Autonomic Behaviors Specifications (ABs) for Selected Diverse Networking Environments (Core Document)”, December 2009
- [2] EFIPSANS project Deliverable D1.5SD “Second Draft of the Specification and Description Tables for Decision Elements (DEs), December 2009
- [3] EFIPSANS project Deliverable D2.3 “Initial Draft Specification of the required Extensions to IPv6 protocols”, June 2009
- [4] EFIPSANS project Deliverable D2.4 “Selected Feature Combination Scenarios for Demo”, August 2009
- [5] OpenVPN online documentation, <http://openvpn.net/index.php/open-source/documentation.html>
- [6] Kunihiro Ishiguro, et al. “Quagga - a routing software package for TCP/IP networks”, July 2006
- [7] Ericsson SmartEdge 800 product page, <http://www.ericsson.com/ourportfolio/network-areas/se800>
- [8] EFIPSANS project Deliverable D2.2: “Specification of the exploitable existing features in IPv6 protocols, including some feature combination scenarios for engineering autonomicity”, December 2008
- [9] EFIPSANS project Deliverable D1.3: “Second Draft of Autonomic Behaviors Specifications (ABs) for Selected Diverse Networking Environments”, December 2008
- [10] EFIPSANS project Deliverable D2.1: “Advanced Routing Algorithms for Autonomic IPv6 Networks”, December 2008
- [11] Bandwidth Test Controller (BWCTL) <http://www.internet2.edu/performance/bwctl/>
- [12] One-Way Ping (OWAMP) <http://www.internet2.edu/performance/owamp/>
- [13] Stoica I, Morris, Liben-Nowell D.; Karger, D.R.; Kaashoek, M.F.; Dabek, F.; Balakrishnan, H., "Chord: a scalable peer-to-peer lookup protocol for Internet applications," IEEE/ACM Transactions on Networking, vol.11, no.1, pp. 17-32, Feb 2003
- [14] EFIPSANS Project Deliverable D.4.3: “Autonomic Networks towards operators, network managers and end users”
- [15] A. Davy et al., Revenue Optimized IPTV Admission Control using Empirical Effective Bandwidth Estimation, IEEE Transactions on Broadcasting, vol. 54, Issue 3, Part 2, pp 599 - 611, Sept. 2008
- [16] MATLAB Simulation & Emulation Toolset, <http://www.mathworks.com/>
- [17] NAGIOS – Industrial Standard in Network Monitoring, <http://www.nagios.org/>

APPENDIX A– EFIPSANS SCENARIOS

As WP5 is only the coordinator of the demonstrations contributed by WP2, WP3 and WP4, the requirements and possible scenarios are defined by other WPs. The initial step of WP5 involves the collection of possible scenarios and the identification of requirements determined by potential trials/demos/scenarios. WP5 is also responsible for an integrated testbed environment, where more complex scenarios are integrated into some trial cases requiring the integration and deployment of all software components resulting from different Frameworks designed in EFIPSANS.

A-1.1 Basic Autonomic Networking Scenarios (WP2)

This section presents the Scenarios targeted for the demonstration of basic networking services such as Discovery, Configuration, Routing and Forwarding. Along each Scenario, we also described the benefits brought by the Technology (Self-* functions) introduced, highlighting the problems/limitations with current technologies, approaches and practices related to network commissioning/provisioning, management and maintenance. As such, we demonstrate how Self-* functions solve the problems described in each Scenario.

The following are the particular scenarios described according to their associated categories of network functionality type as well as the network environment considered for the Scenario.

- Auto-Discovery of Capabilities of Devices and Resources in a Fixed Network Environment.
- Auto-Configuration of Routers using Routing Profiles in a Fixed Network Environment.
- Auto-Configuration/Self-Configuration of Addresses in a SOHO Network:
- The Autonomic DHCP Architecture.
- Auto-configuration/Self-Configuration in MARSIAN (Multi-Hop Mobile Ad-hoc Network Environment).
- Auto-Configuration of radio channels in 802.11 networks.
- Autonomic multi-path routing in 802.11 mesh networks.
- Autonomic Routing in Fixed Network Environments: Self-Adaptation of Routing as driven by Risk-Level Assessment in a Fixed Network Environment.

At the end of each scenario the key features are highlighted, including IPv6 and IPv6++ features that collectively enable the realization of the Scenario—while requiring that the underlying network is an IPv6 network as is expected by the objectives of EFIPSANS.

A-1.1.1 Auto-Discovery and Auto-Configuration Scenarios

A-1.1.1.1 Self-Configuration of Routers using Routing Profiles in Fixed Network Environment

The Auto-Discovery and Auto-Configuration (Self-configuration) scenario presented here below is a composition of the Auto/Self-Configuration scenarios captured in D-2.2 [8]. The “big-picture” scenario presented here aggregates the individual scenarios of D-2.2 [8] and provides a unified and matured view of the Auto-Discovery and Auto-Configuration process during the provisioning of a GANA network. The sections below provide both a high-level and technical description of this unified scenario.

High-Level Description of the Scenario

Auto-Discovery and Auto-Configuration of Routers using Routing Profiles in a Fixed Network Environment	
The Story-line	<p>The network operator, network provider, or an enterprise, an ISP or simply, whichever actor/player who wants to build a network from scratch, has just purchased GANA conformant routers and has been advised by the manufacturer (consults the device manual) as to what minimal manual settings he/she needs to configure on the new router before initializing the router on a link. What he has learnt also, is that he/she need not worry any more about the following (as is used to be the case today):</p> <ul style="list-style-type: none"> - Thinking of dividing and when to divide his/her network into Routing Areas while taking into account the Capabilities of routing devices. - Sitting in front of each router, configuring the router by activating a configuration profile via the Command-Line Interface (CLI). - He/she has been told that from the moment he/she builds the network from scratch, provided that he/she deploys first the ONIX and the required Network-Level-Decision-Elements (NET_LEVEL_DEs), he/she only needs to do the minimal manual settings to initialize a router on a link, since the GANA network would start automatically taking care of fully configuring the routers into service as well as network partitioning as the network grows.
Short description of the scenario	This scenario demonstrates the auto-discovery and auto-configuration processes of network devices from the bootstrapping of the devices in the network to the area partitioning of the routers in the network.
Scenario scenes	<p>The scenario is divided into the following scenes.</p> <ul style="list-style-type: none"> - Scene 1: When the First Node/Device is attached and initialized on a link. - Scene 2: When more Nodes/Devices join the Network. - Scene 3: A special case when a new node initiates a request for a Node-Level Sub-Profile with the NET_LEVEL_RM_DE.
Current problems with current	With the current practices and technology, a network operator / administrator is

practices and current technology	burdened in the following ways. <ul style="list-style-type: none"> • An operator is required to create a configuration profile for each router and at the same time is required to envision the future partitioning of the network into various routing areas. This is a tedious task and is both expensive and prone to errors. • While some features of auto-discovery are available in today's network, they need to be augmented towards more advanced Auto-Discovery as being proposed by EFIPSANS, which includes discovery of Capabilities of devices that plug into the network. For instance, in today's network, OSPF lack box-level discovery and self-description. This is a problem, as the operator has to manually assign an interface for OSPF. 	
Network Environment	Fixed/Wired	
Self* Functionalities introduced	Self-*Functionality	Problems/Limitations it addresses
	Self-Configuration	<p>Current Practices</p> <p>With the current network management practices, a network administrator would need to initialize a router on a link after purchase, and manually create a configuration profile for the router while at the same time thinking of how to assign the router to a routing area (talking about large ISPs) when taking care of network expansion and the benefits of dividing a network into areas.</p> <p>Current Technology</p> <p>A) With the current technology that comes with a router (as manufactured by the manufacturer), the device can only activate factory settings upon initialization on a link(s) while requiring the operator to manually configure its interfaces and routing protocols.</p> <p>B) On the other hand, with the current network management technologies, though some limited remote configuration management commands/data can be issued/communicated to a router, this can only be done after a human has done some initial configuration of the device e.g. IP interfaces (especially the management interface) have been configured. The configuration commands or data do not cover complete or partially complete configuration profiles for the device.</p>

	<i>Auto-Discovery</i>	<p><u>Current Technology</u></p> <p>The current technology that comes with a router does not enable the router to discover entities (Network-Level-DEs) in the network that are tasked with providing each routing node/device with a Configuration Profile that tells it the role the router should play (for example, the router could play the role of an access/edge, core or area-boarder router, etc) as well as the interfaces for which it should configure some routing protocols and their parameters. Though IPv6 brings some aspects of neighbour discovery allowing on-link neighbouring nodes to discover each other and some basic parameters for auto-configuration, these enabling IPv6 mechanisms are still limited and need to be augmented towards more advanced Auto-Discovery as being proposed by EFIPSANS, to even include discovery of Capabilities of devices that are plugged into the network.</p>
	<i>Self-Description and Self-Advertisement</i>	<p><u>Current Technology</u></p> <p>The current technology that comes with a router does not enable the router to <i>self-describe</i> its <i>Capabilities</i> as a device in order to then <i>self-advertise</i> the <i>Capabilities Description Model</i> to the entities that need such vital knowledge such as the Network-Level-DEs that require such knowledge in order to assign a role for the newly attached router and to give it a Configuration Profile which must be used by the router to <i>self-configure</i>.</p>
Self-Configuration – What it solves and the benefits	<p>Owing to the Network-Level-DEs, including the <i>Security-Management DE</i>, the <i>Function-Level Routing-Management DE</i> that comes with a GANA conformant router, as well as ONIX (which supports secure storage and dissemination of information e.g. Configuration Profiles, and enables nodes/devices to discover such things as services, nodes/devices of desired capabilities, etc), <i>more advanced Auto/Self-Configuration than available through IPv6 can be achieved</i>. Self-configuration solves the problem of eliminating the need for a human to manually configure each router with a complete Configuration Profile while at the same time having to think of assigning a router to a routing area in situations where dividing the network into areas is needed.</p>	
Auto-Discovery – What it solves and the benefits	<p>A node initialized on a link discovers ONIX to first learn about how to locate Network-Level-DEs in order to get a role assigned and to finally get an assigned Configuration Profile from ONIX for use in self-configuration. Network-Level-Routing-Management-DEs discover through ONIX; the Capabilities of nodes/devices being attached to the network and use the knowledge in order to assign a role for the newly attached router and to give it a Configuration Profile which must be used by the router to self-configure. The human does not need to think about assigning roles and Configuration Profiles.</p>	

Self-Description and Self-Advertisement – What it solves and the benefits	Self-description of Capabilities by a device and the self-advertisement of the Capabilities Description Model into the ONIX enable Network-Level-DEs assign a role to a device based on its Capabilities.	
System(s) Involved	<ul style="list-style-type: none"> - Core Routers - Edge Routers - Access Routers <p><i>Note: Each of these devices must have been designed to conform to GANA and therefore must embed, apart from the <code>NODE_MAIN_DE</code>, the Function-Level Routing-Management Decision-Element (<code>FUNC_LEVEL_RM_DE</code>)</i></p>	
Key players that benefit	Player	Benefits
	Operator	Benefits at boot up time Reduces OPEX by minimizing the time and effort requirements needed for configuring network devices by the network management personnel (i.e. administrators).
	Manufacturer	Need not explicitly provide software tools for configuration of the network devices and by designing GANA conformant devices, interoperability with other GANA conformant devices is guaranteed.
	End User	N/A

Detailed Technical Description of the Scenario

Profiles:

A draft description of the *Profiles* can be found in D-2.2 [8]. A *Network Profile* in the context of EFIPSANS and GANA is defined as a detailed, structured and monolithic composition of all the information required to configure the network and realise its network goals and objectives. Thus a *Network Profile* is a direct translation of a set of textual business goals of a network into sophisticated and structured technical goals of the network. A *Network Profile* can thus be considered as composition of *Policies*, *Objectives*, *Topology Information* and *Configuration Data*. The components of a *Network Profile* can be thought to be structured in two ways, *Vertical* or *Hierarchical Composition* and *Horizontal* or *Functional Composition*. Detailed information regarding the *Vertical* and *Horizontal Compositions* can be found in section 3.1.1.1 of D-2.2 [8].

In brief, a *Vertical* or *Hierarchical Composition* can be considered as a structuring of the Profile components (namely *Policies*, *Objectives*, *Topology Information* and *Configuration Data*) along the hierarchical levels of GANA. This means that the concept of a profile can be associated with any functional entity from the network, down to a node/device and down to even an individual protocol. A *Profile* encapsulated by a higher level profile is called a *Sub-Profile* of the encapsulating profile. Thus there are four *Hierarchical Profiles* (the lower-level *Profile* being a *Sub-Profile* of the upper-level *Profile*), namely:

- Network-Level Profile
- Node-Level Profile
- Function-Level Profile
- Protocol-Level Profile

Each *Network-Level Profile* can consist of many *Node-Level Profiles* and each *Node-Level Profile* can have many *Function-Level Profiles* and so on.

A *Horizontal* or *Functional Profile* can be considered as the structuring of the Profile components along the abstracted functionalities of GANA, namely, Routing, Forwarding, QoS, etc.

Capability Aggregation:

When a GANA node boots up in the network, the NODE_MAIN_DE initializes itself and bootstraps the underlying DEs and protocols with the minimum configuration parameters provided in the factory settings of the device. Thereafter it initiates the aggregation of Capabilities (*Capability Description Model - CDM*²) of the underlying DEs, MEs and protocols as shown briefly in Figure 18.

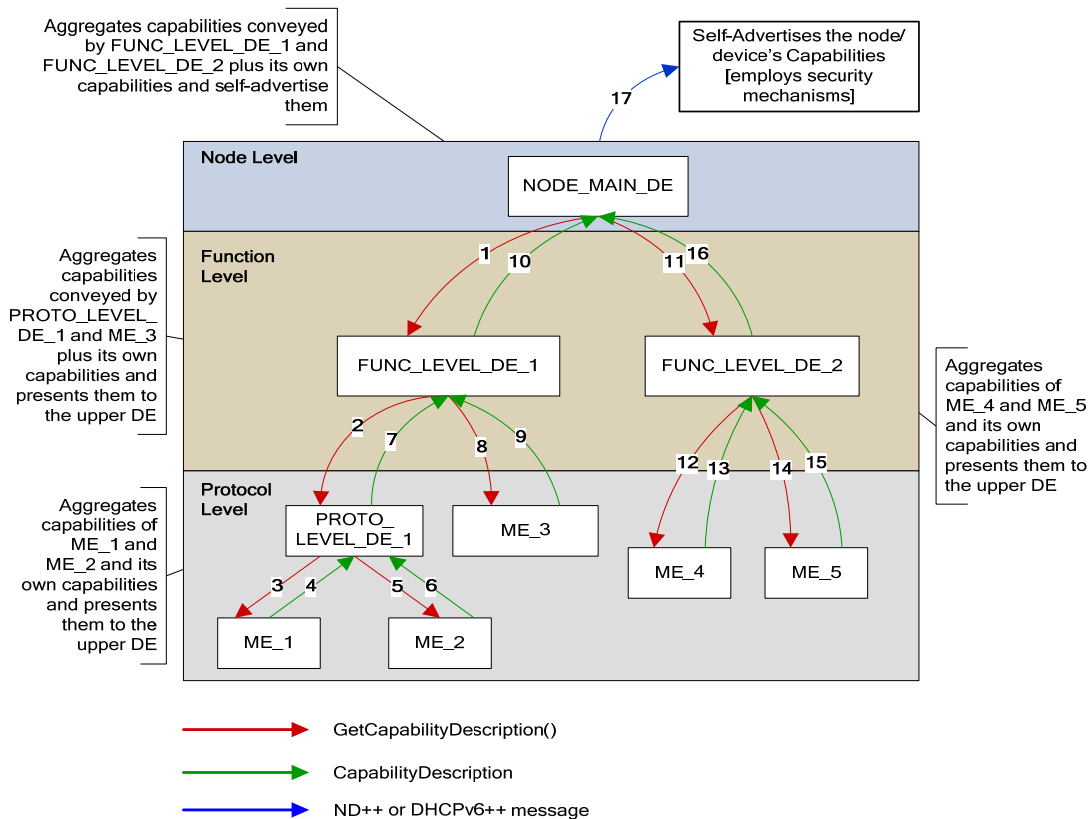


Figure 18: Aggregation of Capabilities of a Node by the NODE_MAIN_DE

² A detailed description of the *Capability Description Model (CDM)* can be found in section 2.1.1.2 of D2.2

Summary of the key Features-Combinations that enable the realization of the Scenario

Scenario Name	Auto-Discovery and Auto-Configuration of Routers using Routing Profiles in a Fixed Network Environment	
Protocol	Summary of the Features that enable the realization of this Scenario	Decision Element(s) involved and the Self-* functionality(ies)
IPv6 core protocols	Stateless address auto-configuration or the use of some more advanced address configuration methods such as the one proposed for the Autonomic DHCP architecture applied to a SOHO network.	NODE_LEVEL_AC_DE (self-configuration)
DHCPv6++	<ul style="list-style-type: none"> - Discover ONIX system - Use services provided by ONIX system (discover devices, services, information; subscribe for receiving updates, etc) 	<ul style="list-style-type: none"> - NODE_MAIN_DE - NET_LEVEL_RM_DE - NET_LEVEL_QoS_M_DE (self-description, self-advertisement, auto-discovery)
ICMPv6++	<ul style="list-style-type: none"> - Enables DE-DE communication. - Is used for the dissemination of the Profiles, and GANA Tokens. 	<ul style="list-style-type: none"> - NODE_MAIN_DE - NET_LEVEL_RM_DE - NET_LEVEL_SM_DE
Other Types of enabling Functional Features	In this scenario the emphasis is on auto-configuration at boot time. For a complete demonstration of the more complex self-* functionalities such as self-adaptation and self-optimization (in an extended version of this scenario) that is based on this scenario as the base scenario, we assume that monitoring should be in place. For the purpose of demonstrating this auto-configuration scenario monitoring is not required.	N/A

A-1.1.1.2 Auto-Collaboration for Optimal Network Resource Utilization in a Fixed Networks

This scenario is aimed at presenting selected features of the ONIX system and how ONIX plays a key role in the process of enabling the collaboration among Content Providers and the serving Network Provider in effecting optimal network resource utilization. ONIX system is described in detail in [8]. We demonstrate how some basic self-* functionalities like self-description and self-advertisement can be exploited in order to enable some more complex self-* functionalities like self-adaptation and self-optimization with the main emphasises on the required “*Content-Provider* \leftrightarrow *Network Provider*” collaboration processes that take place during the operational time of the network.

High-Level Description of the Scenario

Auto-Collaboration for Optimal Network Resource Utilization in a Fixed Network Environment	
The Story-line	<p>Four people, Alice, Bob, Charlie and Dave, unknown to each other, are returning home from work (at around 5pm) and want to relax while watching their preferred TV shows for half an hour. They can choose from different high definition TV shows provided by two different content providers (CP1 and CP2). They have different preferences and usually switch from one show to another. This is a Video on Demand type of scenario where each user may select what they want, when they want to see it (similar to YouTube), unlike the IPTV type of scenario in which the ISP receives all TV signal at one point and it distributes all the signals (channel) to each Access Centre (similar to T-Home service). This behaviour makes life difficult for the network operator as it cannot build a multicast tree as the content is diverse and it is hard to predict for how long a user will be interested in one show before changing to another. The network operator thus encounters serious difficulties in delivering the content around this peak time (5 pm). Due to budget restrictions that prohibit the network operator from increasing the capacity of network, and as bottlenecks keep appearing during the peak time due to the traffic generated by the two content providers affecting the other services using the same network, the operator decides to limit the bandwidth for the content delivered from CP1 and CP2 between 5pm and 6pm. Alice, Bob, Charlie and Dave begin to complain to the content providers because the quality decreases and they are not able to watch the shows in HD anymore. The content providers make an internal investigation and conclude that their infrastructure is used only 50% at the peak hour and after further investigations they discover that their network operator has limited their bandwidth. CP1 and CP2 complain to the network operator and even threaten to take legal actions if the problem is not resolved immediately. The network operator faces a difficult problem as it cannot invest to extend its infrastructure and at the same time increase the bandwidth for the content providers. Confronted with the problem of losing clients and legal problems, the network operator assigns an internal task force to address this problem. An engineer working on the problem proposes a solution that does not require changing the infrastructure. He informs his manager to try the new and easy to deploy GANA architecture in the network. He shows that if all the routers in the network are upgraded to be GANA compliant (only software upgrade needed) and only some small numbers of new devices are bought (network level DEs) the</p>

	<p>problem could be solved. The manager is impressed by this low cost solution and decides to implement it. The network operator upgrades all the routers' software in the network to be GANA compliant; activates ONIX system in the network and buys the required network level DEs. This upgrade is done without any disturbance to the services currently provided. The results are immediate and all the end users are pleased with the service quality offered by the network operator. The network operator retains its clientele and reputation. The technical description of this scenario starts from the point after the software upgrade of the devices to GANA conformant devices is finished and all the other new GANA devices (ONIX and Network Level DEs) are in place.</p>	
Short description of the scenario	<p>This scenario illustrates the key novelties and functionalities (described in detail below) introduced by the enhanced auto-discovery mechanisms proposed by EFIPSANS. This scenario demonstrates how some basic self-* functionalities like self-description and self-advertisement can be exploited in order to enable some more complex self-* functionalities like self-adaptation and self-optimization. Further it reveals the emerging benefits brought by the type of auto-collaboration advocated in this scenario, for both the network operator and for different types of end users like content providers and content consumers.</p>	
Scenario scenes	<p>The auto-discovery scenario is divided into three scenes:</p> <ul style="list-style-type: none"> - Resource discovery: Self-description, self-advertisement and support for solicitation for capabilities description as the basic functionalities required for enabling enhanced resource discovery. - Service discovery: Service advertisement and complex queries for service discovery. - Self-adaptation and self-optimization based on the Ingress_Egress_Choice_Metric computed using auto-discovery functionalities. 	
Current problems with current practices and current technology	<p>Following the current discovery standards and/or current practices, the network operator has a very limited number of possible solutions, but none with acceptable results for the end users. It cannot use multicast protocols as the content is very diverse and users' behaviour hard to predict. There is no interface defined that can be used to exchange information in a standardized way between ISP and CPs regarding the load in the network. The ISP does not intend to provide sensitive information about its network topology or congestion points to the CPs unless this is presented in an abstract manner and using a standardized interface.</p>	
Network Environment	Fixed	
Self* Functionalities introduced	Self-*Functionality	Problems/Limitations it addresses
	Self-Description & Self-Advertisement	The current technology does not enable the possibility to self-describe and then self-advertise the entire set of capabilities (see [8] for more details about the types of capabilities) of a device using a general and extensible information model.
	Auto-Discovery	The current technology that comes within a device does not enable the device to discover devices with different capabilities (e.g.: Network-Level-DEs) inside the

		network. Moreover, using current technologies it cannot discover capabilities unless it knows the exact information model used for describing those capabilities.
Self-Description and Self-Advertisement – What it solves and the benefits	<i>Self-Description</i> of <i>Capabilities</i> by a device and the <i>Self-Advertisement</i> of the <i>Capabilities Description Model</i> into the ONIX enable Network-Level-DEs or other actors attached to the network to perform self-adaptation and self-optimization. Content providers self-describe and self-advertise their services into the ONIX system. They use the same protocols for advertising services as for advertising hardware capabilities.	
Auto-Discovery – What it solves and the benefits	Network-Level-QoS-Management-DE discovers, through the ONIX, the Capabilities of nodes/devices being attached to the network and also the utilization of resources from the network. It uses this knowledge about Capabilities (both static and dynamic [8]) to compute different metrics. Content providers discover the capabilities of the ISP network (e.g.: the possibility to compute load metrics). Content Providers discover, through the ONIX, information about the load in the ISP network and can deliver better QoS to the end users. The information shared between ISP and CPs is stored into the ONIX at an abstract level so that the CPs gains no knowledge about the ISP network topology or congestion points, but this information is still useful for better QoS. Home users devices discover services offered by the CPs using the same protocols as for advertising or discovering hardware capabilities.	
System(s) Involved	<ol style="list-style-type: none"> 1. An operator running a network (ISP network) that is composed of three Core Routers and four Border Routers. 2. Two types of end-users: <ol style="list-style-type: none"> a. Two Content Providers (CP1 and CP2) that use the ISP network to deliver media streams (high definition TV shows) to their clients. CP1 has two locations (CP1A and CP1B) from where it can deliver content for load balancing purposes and CP2 delivers content only from one location. b. Four users (Alice, Bob, Charlie and Dave) who are interested in receiving the content provided by CP1 and CP2. 3. ONIX system 4. NET_LEVEL_RM_DE, NET_LEVEL_QoS_M_DE. <p><i>Note: Each of these network devices (routers) must have been designed to conform to GANA and therefore must embed, apart from the NODE_MAIN_DE, the Function-Level Routing-Management Decision-Element (FUNC_LEVEL_RM_DE) and the Function-Level QoS-Management Decision-Element (FUNC_LEVEL_QoS_M_DE)</i></p>	
Key players that benefit	Player	Benefits
	Operator	Benefits during operational time <ul style="list-style-type: none"> - Possibility to provide load information in an abstract manner. - Optimal utilization of the network resources under given QoS constraints even at peak hours. - Overall load balancing of the whole network.

		<ul style="list-style-type: none"> - Revenue maximization <p><i>Note: Benefits at boot up time are addressed in the Auto-Configuration scenario</i></p>
	Manufacturer	N/A
	End User	<ul style="list-style-type: none"> - Content providers manage to provide the required quality and increase their revenue. - Home user receives the desired quality for their services.

Detailed Technical Description of the Scenario

In this scenario we want to demonstrate the benefits brought by enhanced auto-discovery mechanism proposed by the EFIPSANS project. The description bellow starts from the point after the software upgrade of the devices to GANA conformant devices is finished and all the other new GANA devices (ONIX and Network Level DEs) are in place. When each device boots up into the network it constructs and advertises its capabilities to the ONIX system. The Network-Level-DEs may use the capabilities (both static and dynamic) and other information for computing different metrics. Content providers and ISP may use these metrics for taking decisions to improve the quality of service and the overall utilization of the network resources. The first part of the scenario will focus on presenting the process of auto-discovery and on how the ONIX system facilitates this process and the building of the network topology view by the NET_LEVEL_QoS_DE. The second part of the scenario will focus on how the information collected during the auto-discovery process can be used to influence the decisions of different players (e.g.: NET_LEVEL_QoS_DE, NET_LEVEL_RM_DE, Content Providers) for providing better network services. In the second part we will highlight the role of ONIX as a disseminating system and how the NET_LEVEL_RM_DE and the NET_LEVEL_QoS_DE can collaborate in order to better utilize all the network resources.

Scene 1: Resource discovery: Self-description, self-advertisement and support for solicitation for capabilities description as the basic functionalities required for enabling enhanced resource discovery.

In the first scene of the scenario we start with the configuration presented in Figure 19. We then seek to show what happens when we add a new router and new links connected to this router. The new router is denoted as CR3 and the four new links are between CR3-CR1, CR3-CR2, CR3-BR3 and CR3-BR4 (see Figure 20).

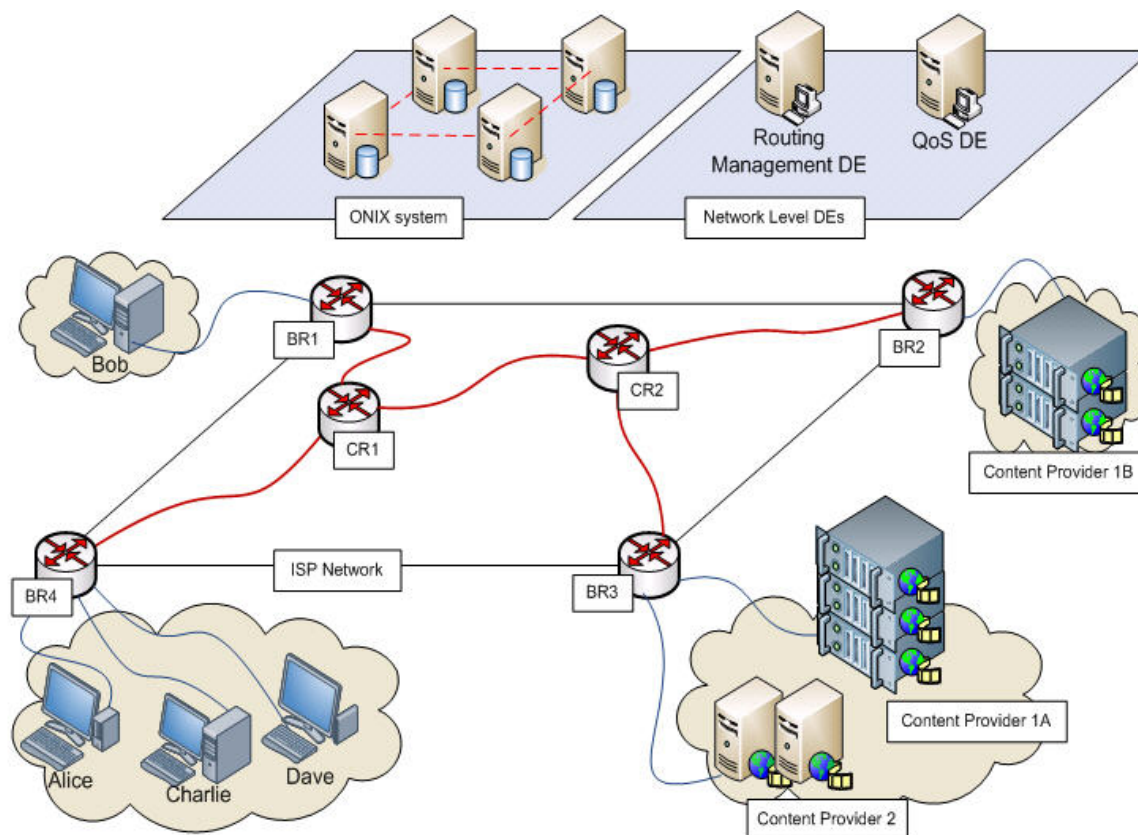


Figure 19: Initial configuration

During CR2 boot up the NODE_MAIN_DE initializes itself and performs the minimum configuration of its secure Neighbour Discovery (ND) related parameters and aggregates the device/node's Capabilities Description Model (CDM). It then tries to discover the ONIX system of the network and it publishes its capabilities into the ONIX. The NET_LEVEL_RM_DE receives the capabilities of the new router from the ONIX system and then it analyses the CDM of the node, and decides the type of "Node-Level Routing Profile" the node would get from the ONIX. Using a *Key* that encapsulates a *Permitted ONIX Operation Token* of type 0 (On-Behalf Subscription), it subscribes the Node to receive a certain type of "Node-Level Routing Profile".

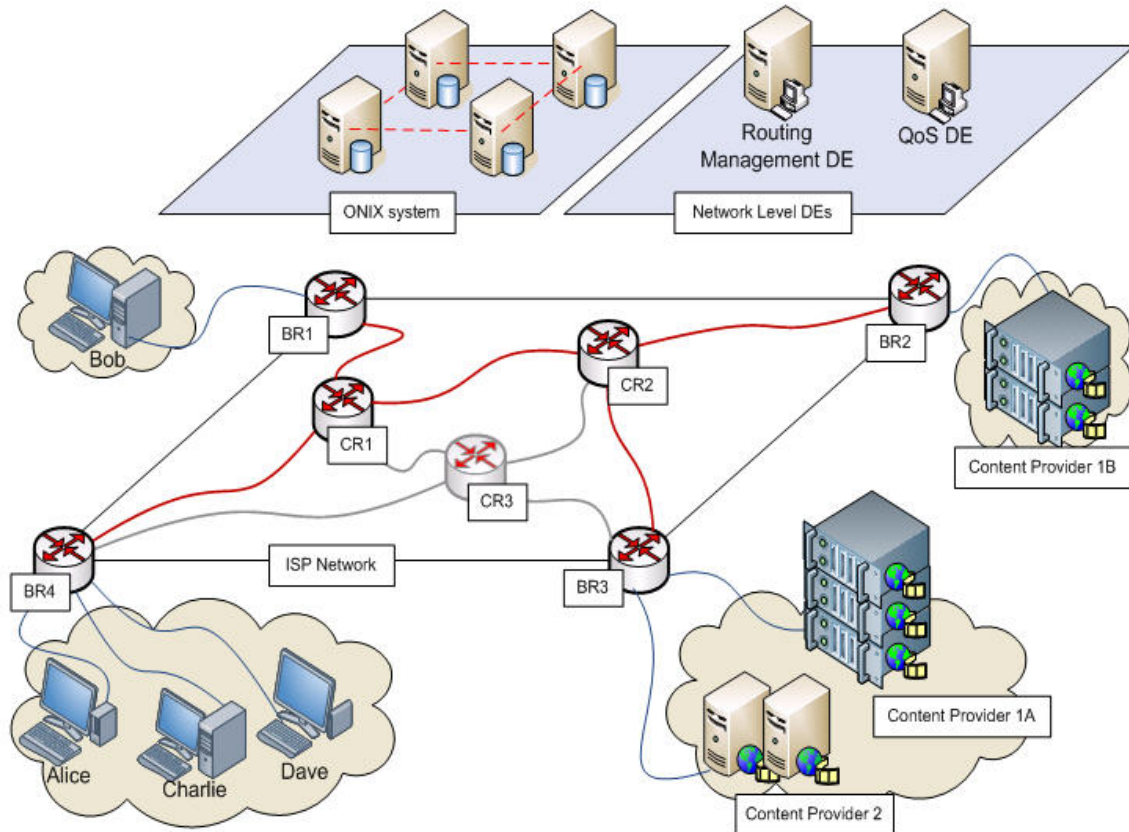


Figure 20: A New Router Is Added Into the Network

Not only the new node that joins the network, but also a number of other nodes already in the network, have to configure themselves, as it may be necessary to perform some auto-configuration for the new links that come with the new router. In our example CR1, CR2, BR3 and BR4 have to configure the new interfaces. The details about how to configure the new interfaces are included in the routing profiles received from the ONIX system during the auto-configuration process. If the NET_LEVEL_RM_DE decides based on the new network architecture that some devices in the network should play a different role, it will update the *on-behalf subscriptions* and the nodes will receive the new profiles corresponding to the new roles. When the new node and all the other nodes finish the configuration of the new interfaces and all the links are ready, all nodes involved will update their capabilities, including the routable IP addresses of each new interface in the updated capability description. When ONIX then receives this information it notifies NET_LEVEL_QoS_DE. This DE will be able to update the values for the Ingress_Egress_Choice_Metric (see Table 4). At some point in time a new software update may be available for the new device in the network, allowing it to filter traffic, for example. The administrator connects to the machine using a CLI and run a script for activating the new software module. After the new module is activated, NODE_MAIN_DE will update the capabilities description into the ONIX system for reflecting the new traffic filtering properties. ONIX then notifies NET_LEVEL_QoS_DE.

Note: For the purpose of the demo, the NET_LEVEL_QoS_DE will have a management console with a graphical interface where the network topology will be displayed (similar to what we can see in Figure 19 or Figure 20). It will be possible to click on different elements composing the network (e.g. devices, links, etc.) and a box will be displayed showing the capabilities of the selected element. When changes appear in the network they are highlighted in the network topology

view. For example: When new links are available but not yet configurable they will appear grey in the topology. After the configuration they become green. When a node updates its capabilities an icon can appear for several seconds next to it.

Scene 2 & 3: Using the Ingress Egress Choice Metric for providing QoS

CP1 is using a load balancing application for better distribution of the load between its two locations and for providing better services to its clients. The load balancing application is taking into account two metrics for deciding which location should deliver content to which user. The first metric is computed by the Content Provider based on the load of the servers (e.g.: CPU used). The second metric is provided by the ISP and it is based on routing information (e.g. the “distance” from the user being served to the location from where the content is delivered) and also on information about the load in the network. Let’s call the first metric CP_Load_Metric and the second one Ingress_Egress_Choice_Metric. CP_Load_Metric takes values in [1, 5], with 1 meaning that there is no load on a server and 5 meaning that the server is 100% loaded. The ISP is not willing to share information regarding the network load and the points of congestions in the network to the CP1 as these are sensitive information that can affect its business. But because of the enhanced auto-discovery mechanism existing in GANA networks, the ISP has a complete view of its network and the points of attachment of CP1 (both CP1A and CP1B) and CP2 to the network. The NET_LEVEL_QoS_DE can then compute the second metric for each point of attachment of content providers to each border router in the ISP network and store this metric inside ONIX. This metric takes values in [1, 5], with 1 being the best choice and 5 the worst. For the network presented in Figure 20 the Network Level QoS DE will compute this metric for requests having the ingress point BR1 or BR2 and the egress point BR3 or BR2 and it will periodically update the values stored in ONIX for each of the following pairs: BR1-BR2, BR1-BR3, BR4-BR2 and BR4-BR3.

This metric is computed from time to time, not in real-time. CP1 can use this metric to decide for a specific user from which location to deliver the requested content. CP1 will not interrogate ONIX every time a user issues a request for content, but, instead, it only needs to ask ONIX periodically, or even better it can subscribe to the ONIX system to receive notifications when the value of these metric changes. In this way, the ISP does not share sensitive information about its network load to third parties and at the same time collaborates with the content provider to deliver high quality services to their end users.

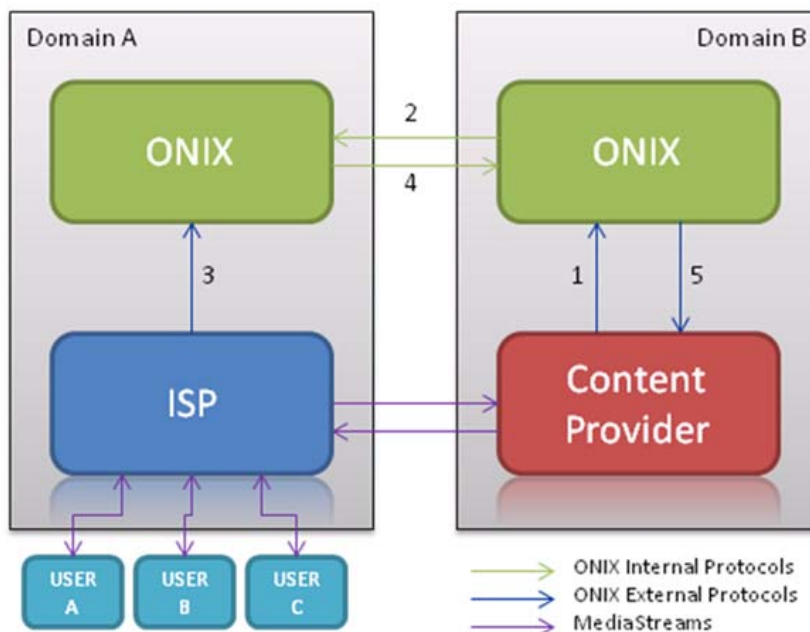


Figure 21: Inter-domain Resource Discovery

The messages exchange necessary for this operation are presented in Figure 21. The CP subscribe to the ONIX system for receiving updates about path load (message 1). ONIX system from Domain B will forward the request to ONIX system from Domain A (message 2). When the load from the ISP network changes ONIX system from Domain A will be informed (message 3). Since it has a subscription for this type of information it will forward the changes to ONIX system from Domain B (message 4). Finally the updated information will be received by the CP (message 5). The steps necessary for the CP to discover the Ingress_Egress_Choice_Metric and to subscribe for receiving updates regarding this metric are presented in the next sequence diagram (Figure 22).

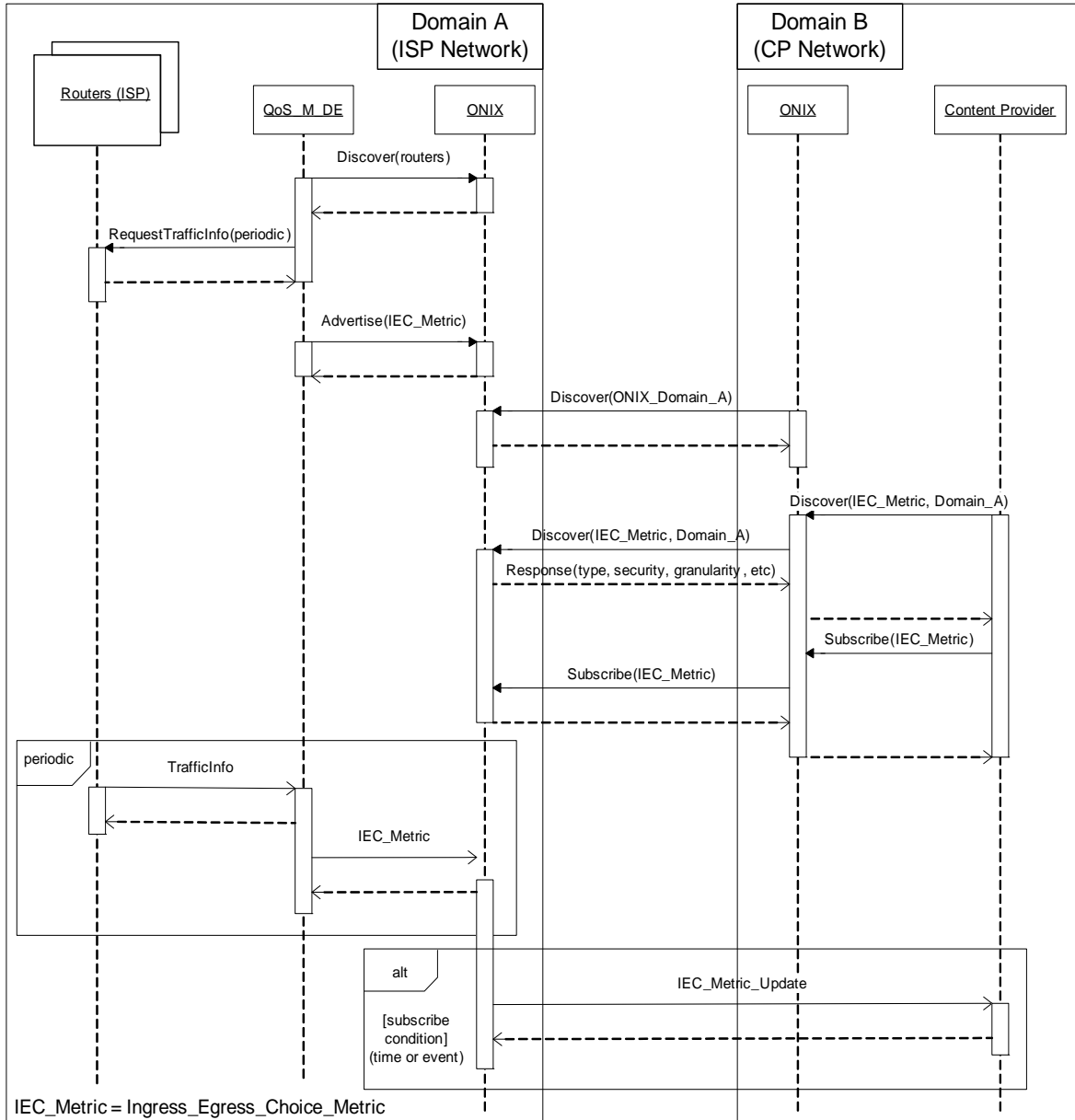


Figure 22: Sequence diagram for using the Ingress Egress Choice Metric

Using the configurations presented in Figure 19 and in Figure 20 we demonstrate the benefits conveyed by auto-discovery and the use of ONIX system in the following sequence of steps. The values for each metric at different steps are presented in Table 7.

- 1) A new router is added into the network and the Ingress_Egress_Choice_Metric changes according to the new configuration.
- 2) Alice requests some content (a HD TV show) from CP1. According to the Ingress_Egress_Choice_Metric Alice terminal will get the content from CP1A. The TV show last for several minutes, at least until the end of the scenario. The same is valid for all other users.

- 3) Bob wants to see a TV show also from CP1 as well. Using the Ingress_Egress_Choice_Metric Bob may receive content from CP1A and CP1B. But since CP1A is already providing content to Alice, for load balancing reasons, taking also into account the CP_Load_Metric the CP1 decides to provide the content requested by Bob from CP1B location.
- 4) Charlie is also interested in watching a TV show, provided by CP2 and not from CP1. CP2 will deliver the content to Charlie terminal.

Dave wants to watch a TV show from CP1. The load balancing application used by CP1 must take into consideration both metrics. The servers load is similar because each of them is serving one user hence the CP_Load_Metric has the same value for CP1A and CP1B. Both Alice and Charlie receive content from CP2. This means that the Ingress_Egress_Choice_Metric will have a greater value for BR4-BR3 than for BR4-BR2. Hence CP1 will decide to deliver content to Dave

Table 7: Metrics values over time

Steps	Ingress_Egress_Choice_Metric				CP_Load_Metric		
	BR1-BR2	BR1-BR3	BR4-BR2	BR4-BR3	CP1A	CP1B	CP2
State Before Step 1 (Figure 19)	2	2	2	2	1	1	1
State After Step 1 (Figure 20)	2	2	2	1	1	1	1
State After Step 2	2	2	2	2	2	1	1
State After Step 3	3	2	2	2	2	2	1
State After Step 4	3	2	2	3	2	2	2
State After Step 5	3	2	3	3	2	3	2

Summary of the key Features-Combinations that enable the realization of the Scenario

Scenario Name	Auto-Collaboration for Optimal Network Resource Utilization in a Fixed Network Environment	
Protocol	Summary of the Features that enable the realization of this Scenario	Decision Element(s) involved and the Self-* functionality(ies)
IPv6 core protocols	Stateless address auto-configuration or the use of some more advanced address configuration methods such as the one proposed for the Autonomic DHCP architecture applied to a SOHO network	NODE_LEVEL_AC_DE (self-configuration)
DHCPv6++	<ul style="list-style-type: none"> - Discover ONIX system - Use services provided by ONIX system (discover devices, services, information; subscribe for receiving updates, etc) 	<ul style="list-style-type: none"> - NODE_MAIN_DE - NET_LEVEL_RM_DE - NET_LEVEL_QoS_M_DE (self-description, self-advertisement, auto-discovery)
Other types of enabling Functional Features	In this scenario the emphasis is on auto-discovery and how it can lead to more complex self-* functionality (e.g. self-adaptation, self-optimization). For a complete demonstration of the more complex self-* functionalities (in an extended version of this scenario) we assume that monitoring should be in place also. For the purpose of demonstrating auto-discovery monitoring is not required in this scenario.	N/A

A-1.1.2 Auto- and Self-Configuration Scenarios

A-1.1.2.1 Auto-Configuration/Self-Configuration of Addresses in a SOHO Network

The scenario described in this section is a special case of the one of sectionas it demonstrates how network devices enriched with GANA functionality can auto-configure their addresses when the network administrator or just an untrained user builds or extends a Small Office/Home Office (SOHO) type of network.

In the SOHO type of networks, network administrators are generally the end-users without technological education or system administrators with limited knowledge or time. They are not quite trained to manually edit configuration files, install software components, etc. Therefore, the auto-configuration capability and, more specifically, the address auto-configuration capability of the network is a key issue for deploying SOHO networks. This requirement was recognized by the standardization bodies as well, so IPv6 from the very beginning offers two standardized IPv6 host address auto-configuration schemes that enable an end-host to auto-configure itself with IP address without human intervention. However, there has not been an easy way to automatically configure an entire SOHO site, mostly because there were no proposals for routers' auto-configuration.

High-Level Description of the Scenario

Auto-Configuration/Self-Configuration of Addresses in a SOHO Network: The Autonomic DHCP Architecture	
The Story-line	We envision the following scenario for demonstrating of the Autonomic DHCP architecture and its auto-configuration capabilities. Let us assume that a system administrator of a university is charged with the task to upgrade the department computer facility with a new laboratory, complete with student terminals, servers, printers, scanners, multimedia devices, etc. He interconnects several computers (routers and end-hosts) over Ethernet, and without much ado each of them will receive a unique, routable IPv6 address from a central DHCP server. Moreover the routers will take up the role of a DHCP relay agent when necessary. He then checks whether the network is operational by reading the status messages of interface cards.
Short description of the scenario	Autonomic auto-configuration of a small SOHO network with unique, subnetted IPv6 addresses using the Autonomic DHCP architecture. Note that the scenario does involve the self-configuration of the routing functionality; we are only concerned with address auto-configuration.
Scenario scenes	<p>The scenario is aimed at demonstrating the basic steps of building/setting up/extending a SOHO network and auto-configuring proper IPv6 addresses, starting from a sole edge/border router, and ending with a network with several routers and hosts. The scenario consists of the following scenes:</p> <ul style="list-style-type: none"> - Router address auto-configuration without relaying (i.e., with direct access to the Main DHCP server) - Router address auto-configuration using the Link Bootstrapping Feature with simple relaying - Router address auto-configuration using both the Link Bootstrapping

	Feature and the Smart Relaying Feature <ul style="list-style-type: none"> - Client address auto-configuration (using standard IPv6 client software) through the Smart Relaying Feature - Detaching and reattaching a link, demonstrating the Evolvability/Flexibility/Extensibility/Robustness (EFER) Feature provided by the Autonomic DHCP architecture. 	
Current problems with current practices and current technology	Following the current standards and technology, the system administrators have no choice but manually configure all the routers in their networks. Besides assigning addresses to some interfaces, this configuration process may include the setup of DHCP Relay Agents, or alternatively the setup of DHCP Servers with proper prefix delegation options. In both cases, however, the burden on the system administrator does not come just from the manual configuration, but the exact topology of the network with the addressing scheme also has to be planned beforehand the installation of the equipments.	
Network Environment	Fixed/Wired	
Self* Functionalities introduced	Self-*Functionality	Problems/Limitations it addresses
	Self-Configuration	Since the scenario fits into the general framework of auto-configuration as envisioned by the EFIPSANS project, the generic limitations of the current practice outlined in Scenario 3.1 apply here as well. More specifically, the concerns formulated in the row Current problems with current practices and current technology column of the present table apply here as well.
Self-Configuration – What it solves and the benefits	Address auto-configuration as described in this Scenario eliminates the need for a human to <ul style="list-style-type: none"> • Manually configure each interface with a unique, routable IPv6 address, • Each router with unique IPv6 address prefixes to advertise on its interfaces for hosts to auto-configure, designing an addressing plan, • And setting up a chain of DHCP relays or servers for more complex networks. 	
System(s) Involved	1. Routers 2. Hosts <i>Note: We assume that there exists a node playing the role of the Main DHCP server, which is configured with a global address pool to assign addresses from for clients. This node may, or may not coincide with the NET_LEVEL_AC_DE.</i>	

Key players that benefit	Player	Benefits
	Operator	The same as for Self-configuration in general: reduces OPEX by minimizing the time and effort requirements needed for configuring network devices.
	Manufacturer	N/A
	End User	Facilitates for an end-user without specific training to administer a SOHO network, as long as address configuration is concerned.

The detailed Technical Description of the Scenario

The demonstration will allow the audience to follow the procedure through which the Autonomic DHCP architecture configures a fixed SOHO network with unique IPv6 addresses. This involves tracking and visualizing the link bootstrapping process, the demonstration of the DHCP smart-relaying functionality, etc. Each scene of the scenario, as described in Table **Auto-Configuration/Self-Configuration of Addresses in a SOHO Network: the Autonomic DHCP architecture**, row **Scenario scenes** correspond to a basic Feature of the Autonomic DHCP architecture, in particular, the Link Bootstrapping Feature, the Smart Relaying Feature and the Evolvability/Flexibility/Extensibility/Robustness (EFER) Feature. For a detailed description of the process model of the Autonomic DHCP architecture, the Features mentioned above and an illustration of the components, DEs, type of nodes, etc. involved in the process, the reader is referred to D-2.3 [3].

The SOHO demonstration network itself will consist of a handful of OpenWrt routers plus some hosts (laptops) attached to some of the interfaces. In the course of the demo, the routers will be connected to a mesh topology (based on wish of the audience), will autonomically acquire proper IPv6 prefixes on their interfaces, and begin to advertise these prefixes to hosts which in turn will pick up suitable unique addresses. For demonstration purposes, an *ssh* connection will be established to one of the routers, whose inner state, state transitions, the steps of the auto-configuration process, events, etc., will be visualized on the laptop's screen.

Summary of the key Features-Combinations that enable the realization of the Scenario

Scenario Name	Auto-Configuration/Self-Configuration of Addresses in a SOHO Network: The Autonomic DHCP Architecture	
Protocol	Summary of the Features that enable the realization of this Scenario	Decision Element(s) involved and the Self-* functionality(ies)
IPv6 core protocols	<ul style="list-style-type: none"> - Router Advertisement mechanism - Stateless address auto-configuration 	NODE_LEVEL_AC_DE
DHCP	<ul style="list-style-type: none"> - Stateful address auto-configuration - DHCP Message relaying 	NODE_LEVEL_AC_DE

DHCP++	Smart relaying Link bootstrapping	NODE_LEVEL_AC_DE
Other types of enabling Functional Features	Self-discovery of status changes of the network interface cards	NODE_MAIN_DE NODE_LEVEL_AC_DE

A-1.1.2.2 Auto-Configuration/Self-Configuration of Addresses in MARSIAN Platform

The scenario for Auto-configuration in MARSIAN is aimed at presenting selected features of MARSIAN platform for fault Management, Auto-configuration, Resilience and Survivability In Autonomic mobile ad hoc Networks with the main emphasis placed on auto-configuration. The scenario considers a multi-hop mobile ad hoc network environment. The scenario aims to present autonomic behaviors such as auto-configuration and potentially also some aspects of self-healing.

High-level description of the Scenario

Auto-configuration for mobile ad hoc networks	
The Story-line	<p>Several mobile nodes, having the auto-configuration features of MARSIAN platform implemented, try to form a multi-domain IPv6 - based ad hoc network. The network is to be self-managing, without the necessity of operator's control. However, a governing entity is usually present in the network. Proposed solutions are beneficial mainly to the end users which do not have to manually configure their devices (or such a configuration is limited to the minimal required level) and can remain connected in the changing multi-domain environments where sub-networks may split, merge and specific faulty conditions might occur. Before the users are able to send user data, network must be properly configured in terms of address auto-configuration. Each node needs to obtain a unique, routable address without manual intervention of the end user. Several restrictions resulting from the MANET environment, such as nodes mobility and constant topology changes must be taken into consideration, so that the user doesn't sense underlying network changes and its connectivity is maintained.</p> <p>Once configured, the network must be able to keep proper configuration regardless of changes in the networking environment. Specifically auto-configuration procedures give support to situations where:</p> <ul style="list-style-type: none"> - New node is joining a network - Sub- networks/domains are merged - Sub- network/domain is split - Node failure emerging at one randomly selected node or at a group of nodes <p>The ability to maintain connectivity within a network, to reach the state where all nodes are properly configured in terms of address auto-configuration, and to offer basic services such as routing, is perceived as a measure of the auto-configuration capabilities and of the level of resilience (and survivability). Additionally it is assumed that the network self-configures taking into account the requirements imposed by the autonomic routing entity. This way there is an interaction which is aimed at providing an optimum network configuration.</p>

Short description of the scenario	<p>Nodes are physically connected by means of wireless 802.11g capable interface and form an ad hoc network in the infrastructure- less mode. The FreeBSD operating system with IPv6 KAME implementation in the system kernel, modified according to the results obtained in EFIPSANS, is controlling the node's behaviour. The multi-hop multi-domain wireless environment is emulated by means of filters set as firewall with configuration specified for each node. All nodes are aimed to form multi-domain environment where nodes remain properly configured even in the case of changes in the network environment. Auto-configuration procedures must be able to adapt nodes' configuration to the current networking situation.</p>
Scenario Scenes	<p>Scenario is divided into several scenes:</p> <ul style="list-style-type: none"> - Node joining a network: auto-configuration of the network in the case where a single new node wants to join a multi-domain networking environment or when a node after a node failure rejoins a network - Network merging: including the situation of auto-configuration actions that need to be introduced while a whole new subnet is joining a networking environment - Network splitting: situation emerging e.g. after moving towards an obstacle or as a result of the lack of connectivity between parts of a domain resulting from other causes, like radio waves propagation disruptions - Node failure emerging at one randomly selected node or at a group of nodes: After detecting such a failure, the network should react properly and be able to auto-configure itself in the new networking situation, providing a proof that auto-configuration features are able to interact with fault management offering improved network resilience and survivability
Current problems/limitations with <i>current practices</i> and/or <i>current technology</i>	<p>The current solutions for auto-configuration in MANETs do not consider combining auto-configuration features with other functionalities implemented in the node (like fault management, resilience and survivability mechanisms), which may be a source of crucial information for making auto-configuration decisions resulting in a better resilience of the system. Auto-configuration solutions based on IPv6 protocol usually do not offer flooding optimization and are not considered fully self-managing approaches. Current practice is that users configure their interfaces manually or addresses are assigned by a DHCP server and after that, the network state changes do not influence the system configuration. The latter might result in service corruption and multiple unnecessary errors while nodes are moving, creating domains and encountering faulty conditions.</p>
Network Environment	<p>Wireless mobile multi- domain ad hoc networks</p>

Self* Functionalities introduced	Self-Functionality	Problems/Limitations it addresses
	Auto-Configuration	<p><u>Current Practices</u></p> <p>Users configure their interfaces manually or addresses are assigned by a DHCP server and after that network state changes do not influence system configuration which might result in service corruption and multiple unnecessary errors while nodes are moving, creating domains and encounter faulty conditions.</p> <p><u>Current Technology</u></p> <p>The current solutions for auto-configuration in MANETs do not consider combining auto-configuration features with other functionalities implemented in the node (like fault management, resilience and survivability mechanisms) which may be a source of crucial information for making auto-configuration decisions resulting in better resilience of the system. Even if some features of IPv6 are involved in the technology, the solution suffers from limitations like not optimal flooding and do not present a general frameworks for addressing auto-configuration in the considered environments.</p>
	Self-Healing (potentially)	<p><u>Current Technology</u></p> <p>The current solutions are mainly concentrating on chosen aspects of a system in order to analyze the selected features of protocols, functionalities, devices, etc. that can be addressed to improve the resilience of this particular type. EFIPSANS aims to show that the benefits emerging from the GANA architecture enable the creation of the general solutions with the aid of which resilient aspects of all the isolated features of a node could be merged and taken into consideration in order to improve the overall resilience and survivability. Potentially, it will be shown that auto-configuration solutions that are made aware of the current node state (with regard to fault management) are able to introduce proper actions to realize the features of a generic resilience architecture.</p>
	Autonomic-Routing	<p><u>Current Technology</u></p> <p>Currently the interaction between auto-configuration, self-healing and routing seems limited. One seeks solutions that would incorporate autonomic routing into MANETs through integration with self-healing and auto-configuration.</p>

		This way the network would be more robust in terms of handling unspecified situations that may appear unexpectedly.
Auto-Configuration – What it solves and the benefits	<p>The Auto- configuration features of MARSIAN platform aim to provide auto-configuration solutions for multi-domain MANETs which would support network merging and splitting in IPv6 - based network. Address assignment along with auto-configuration logic result in keeping proper network configuration in a changing MANET environment. Operations are not only exploiting IPv6 but are also acting in an optimized manner. The end user does not need to manually reconfigure the node's address after each change in the network topology/state/etc. From a higher layer point of view, communication and sessions are sustained regardless of conditions shaping the network's structure.</p> <p>The benefits of auto- configuration are also reflected in the functionality of the MARSIAN platform, where a properly configured network can take advantage of all the benefits resulting from the interactions of auto-configuration, fault management, resilience and survivability, and all its actions can be properly performed.</p>	
Self-Healing (potentially) – What it solves and the benefits	<p>If a network is able to recover from some erroneous state by interacting with auto-configuration features to ensure realization of resilient behaviors by proper network configuration, which results in adaptation to current situation in a network and possible avoidance of future incidents, this becomes a real benefit not only to the end user but also to potential network operator or the network itself.</p> <p>MARSIAN platform aims at supporting proper collaboration of auto-configuration, fault management and resilience in order to provide such a benefit.</p>	
Autonomic-Routing – What it solves and the benefits	<p>When we talk about multiple data streams having different requirements and the network topology is changing dynamically and/or failures occur, the network needs to accommodate for autonomic routing. In that case both the auto-configuration and self-healing engines need to act in such a way that they take into account the requirements imposed by the routing entity. Also observations provided by the monitoring entity are very helpful here. As a result the network might be reconfiguring so the setup is better balanced with regard to specific objectives imposed by policies and the current status.</p>	
System(s) Involved	Mobile nodes having capability to act as host or routers	
Key players that benefit	Actor/Player	Benefits (with Rationale)
	Operator	If a network is managed by the operator only, minimal configuration of the top-level devices, by specifying general configuration profiles, is necessary. The network can then ensure proper auto-configuration in its lifetime, while the domains are merging, splitting and encounter faulty conditions.

	Manufacturer	Devices with the new technology implemented will have capabilities making operation of those devices much easier. In many cases manual configuration will not be necessary and devices will be able to solve many emerging problems without the user's intervention. Such a feature of the device makes it very interesting for a potential customer and would make the device much more competitive. Also, operation and maintenance costs, costs for providing customer help, etc. will be reduced
	End User	A proper device auto-configuration can be assumed without the need for manual configuration after each change in network's structure. The device will be able to deal with many diversified and/or erroneous situations on its own.

The detailed Technical Description of the Scenario

As stated before, the proposed scenario is aimed at presenting selected features of MARSIAN platform for fault Management, Auto-configuration, Resilience and Survivability In Autonomic mobile ad hoc Networks with the main emphasis placed on auto-configuration. The scenario aims to present autonomic behaviors such as auto-configuration and potentially also some aspects of self-healing. The EFIPSANS – proposed solutions for auto-configuration in MANETs will be presented, especially IPv6 extensions described in detail in D-2.3 [3].

An overview of MARSIAN platform is given in D-2.2 [8]. The MARSIAN platform being an integral part of the GANA architecture is able to auto-configure the multi-domain set-up. MARSIAN platform targets especially tactical-like environments where multiple sub-networks can move rapidly and might merge or further split as well as encounter specific faulty conditions. The role of auto-configuration is then not only to enable an efficient address assignment scheme but also provide certain capabilities making it feasible for the network to survive in difficult conditions.

Figure 23 represents GANA architecture for MARSIAN platform. It is important to underline that functional relations between GANA elements are presented, the figure however is not reflecting neither physical nor logical structure of the network; nevertheless some similarities can be observed. In GANA architecture for MARSIAN platform each node is equipped with Node Level Auto-configuration DE, Node Level Fault Management DE and Node Level Resilience and Survivability DE interacting with each other to achieve above specified goals. Auto-configuration DE in each autonomic node steer the auto-configuration process and manages several entities, like ND++_DE, responsible for stateless and stateful address auto-configuration. It is not only responsible for the optimum deployment but also provides quick and efficient IPv6 address configuration so that duplicated addresses can be efficiently avoided. Based on information from Fault Management DE about isolated faults and failures and from Resilience and Survivability DE, Auto-configuration DE can perform the actions necessary to remove or mask faults occurring in the system and therefore increase the level of network's survivability and enhance resilient behaviors. The purpose of this interaction is that on the one hand the AC_DE may try to make decisions including these requirements and on the other hand, even if there are no specific requirements, the interaction between the decision elements might result in a better overall network robustness with regard to the network itself (a group of nodes) and the services it offers.

Several designated nodes are having higher capabilities and are able to configure group of nodes and therefore contain Network Level Auto-configuration DEs. Those DEs possess more information about configuration data and can be evoked in the specially selected nodes. Other Node- or Function- Level DEs may also be instantiated in the designated nodes if such a need occurs. ONIX is envisioned to be used for data dissemination and is able to pass necessary configuration data to Network Level Auto-configuration DEs.

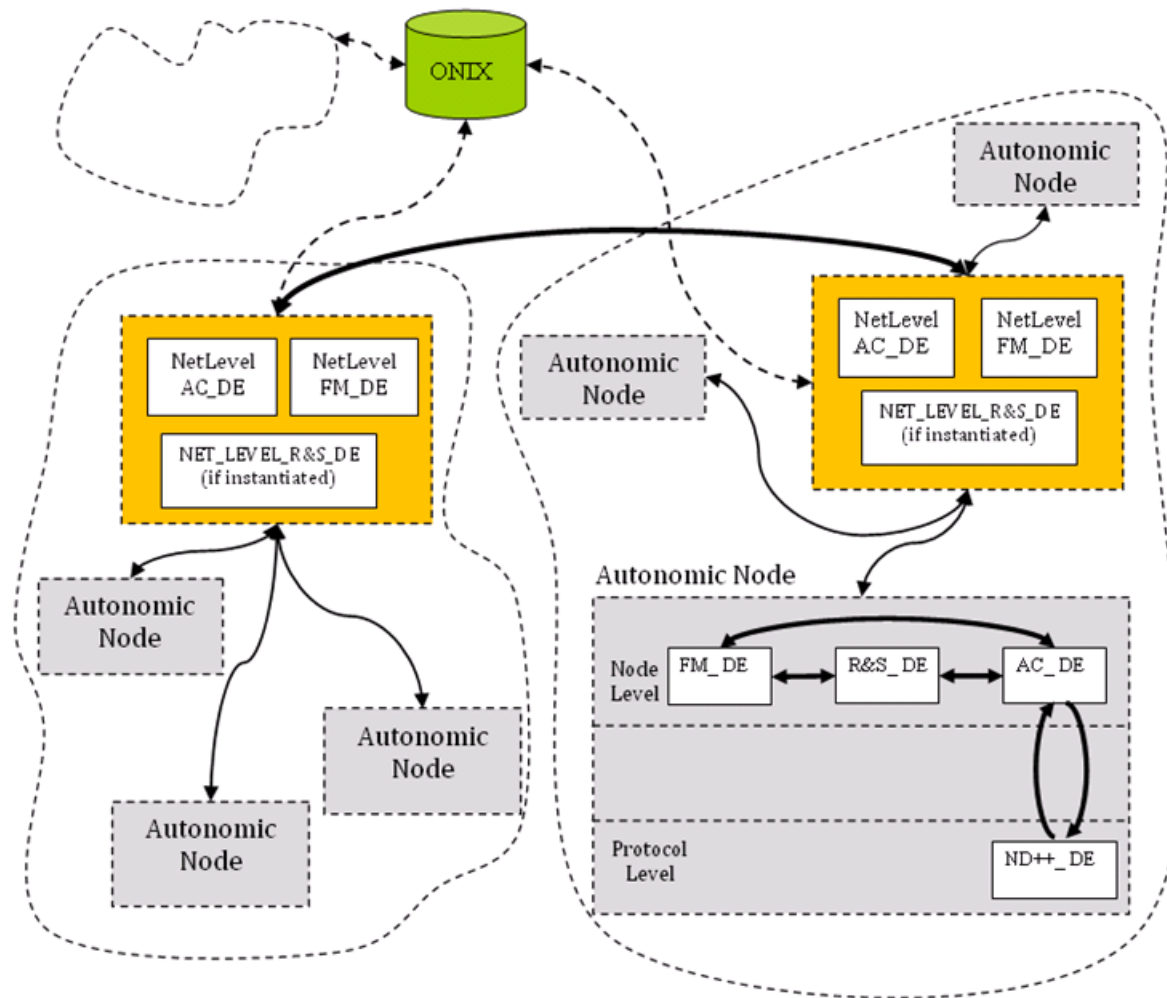


Figure 23: GANA architecture overview for MARSIAN platform

The above described scenario aims at presenting auto-configuration capabilities of the MARSIAN platform. Therefore, selected features of the perceived platform functionality will be presented and possibly links with fault management, resilience and survivability will be shown as a proof of concept. The concept of MARSIAN itself however will not be implemented in full.

The scenario consists of three selected scenes, each of which is presenting different features of the entire scenario:

- **Node joining a network:** auto-configuration of the network in the case where a single new node wants to join a multi-domain networking environment or when a node after a node failure re-joins a

network. New node joining a network starts to perform DAD procedure (as described in detail in D-2.3 [3]) by means of EFIPSANS- proposed extensions to IPv6 ND protocol (ND++) in order to ensure that it will obtain unique IPv6 address within entire domain (and possibly also within entire network formed by a group of domains)

- **Network merging:** including the situation of auto-configuration actions that need to be introduced while a whole new subnet is joining a networking environment. While networks are merged, it is necessary to ensure that each node within a newly created domain will have an IPv6 address whose uniqueness within entire domain can be guaranteed. Proper procedures must be performed which cover not only node- intrinsic functionality, but also designated network elements conveying Network Level Auto- configuration DEs.
- **Network splitting:** situation emerging e.g. after moving towards an obstacle or as a result of the lack of connectivity between parts of a domain resulting from other causes, like radio waves propagation disruptions. In this sub-scenario, presented in Figure 24, one of domains is moving towards an obstacle and is therefore obligated to split into two parts. Those two parts are treated like new sub-networks in the network. In such a case, proper actions must be undertaken, including node- and network- level DEs, so that addresses and prefixes are properly distributed and assigned within a domain in order for the latter to create a fully functional and properly configured entity.
- **Node failure emerging at one randomly selected node or at a group of nodes:** After detecting such a failure, the network should react properly and be able to auto-configure itself in the new networking situation providing a proof that auto-configuration features are able to interact with fault management offering improved network resilience and survivability. Auto-configuration features should ensure that after a node failure, communication between the rest of nodes can be ensured and all addresses are properly assigned and routable.



Figure 24: An example situation covered by the proposed scenario

Summary of the key Features-Combinations that enable the realization of the Scenario

Scenario Name	Auto-configuration for mobile ad hoc networks	
Protocol	Summary of the Features that enable the realization of this Scenario	Decision Element(s) involved and the Self-* functionality(ies)
IPv6 core protocols	Stateless address Auto-Configuration	NODE_LEVEL_AC_DE,
IPv6++	<ul style="list-style-type: none"> - ND++ (EFISPANS proposed Neighbour Discovery ++) - Auto-configuration capabilities of MARSIAN platform 	<ul style="list-style-type: none"> - ND++_DE - NODE_LEVEL_AC_DE - Support from FM_DE, R&S_DE
Other types of enabling Functional Features	Monitoring supporting gathering node-intrinsic information	Support from Monitoring DE (to be implemented in WP4)

A-1.1.2.3 Auto-Configuration of Radio Channels in 802.11 networks

The Auto-Configuration of channels in 802.11 networks is a scenario, which is based on the WARF concept described in D2.2 and is based on the state-of-the art analysis presented in deliverable D2.2 [8]. The WARF framework accommodates cross-layer operations, multiple radio interfaces, real-time resource monitoring, dynamic resource allocation and multi-path adaptive forwarding. WARF is based on commonalities found in existing routing protocols. Thus wireless networks for purposes such as meetings, sports-events or emergency situations can be built based on this framework.

High-Level Description of the Scenario

Auto-Configuration of radio channels in 802.11 networks	
The Story-line	Wireless networks can be built on demand for purposes like meetings, sport events or emergency situations. The typical user of these networks is a nomadic user, i.e. an owner of a portable device equipped with 802.11 interface (notebook, PDA, mobile phone etc.). In such approach typically there is no network operator, but the network is built by the end-users or an owner of the property where the network is installed.
Short description of the scenario	A new network will be installed in an environment with other 802.11 networks under operation. A proper selection of radio channel(s) by a node auto-configuring in the new network will be evaluated. A change of the network configuration (due to the movement of the nodes, change of the transmission parameters) can trigger a channel(s) reallocation.
Current problems with current practices and current technology	In typical wireless networks based on 802.11 devices, there are no built-in mechanisms for proper channel selection, as well in single-hop as in multi-hop (mesh) configurations. The channel selection is made a priori by a network operator or owner. Unfortunately, in the unlicensed ISM, in which these devices are typically (in the 802.11b and 802.11g case), working, there is a limited number of channels: 11 channels in Europe, 13 channels in USA and 14 channels in Japan and some other Asian countries. Moreover the neighbouring channels overlap partially. Due to the enormous popularity of 802.11 devices and other devices working in the ISM band, for example Bluetooth, the probability of radio interferences is very high. Due to this phenomenon the network performance and reliability is degraded.
Network Environment	Wireless, 802.11 based single- and multi-hop networks

Self* Functionalities introduced	Self-*Functionality	Problems/Limitations it addresses
	<i>Self-Configuration of physical layer of 802.11 devices</i>	<p><u>Current Practices:</u></p> <p>At present the manual configuration by a network administrator is only possible.</p> <p><u>Current Technology:</u></p> <p>The SNMP protocol can be used for static and centralized channel allocation</p>
Self-Configuration – What it solves and the benefits	<p>There are two important reasons for autonomic channel selection:</p> <p>The first reason deals with the selection of the best radio channel in terms of interferences, while the second one is related to use of multiple radio interfaces in order to improve the overall network performance. Such an approach is well known in order to overcome the problem of blocking of radio resources due to the packet transmission along a path (the hidden and the exposed terminal problem). Thus, the appropriate channel selection is of great importance in the context of performance and reliability.</p>	
System(s) Involved	<p>Typically these devices are installed as access devices, but in fact they may compose a standalone wireless network.</p> <p><i>Note: Each of these devices must have been designed to conform to GANA and therefore must embed, apart from the NODE_MAIN_DE, the Function-Level Routing-Management Decision-Element (FUNC_LEVEL_RM_DE). The implementation of the presented concepts is relatively easy, due to the fact that some 802.11 devices are working under Linux and the source code is available. So, the implementation of the proposed mechanism is relatively simple.</i></p>	
Key players that benefit	Player	Benefits
	<i>Operator</i>	Reduces OPEX by minimizing the time and effort requirements needed for configuring network devices by the network management personnel (i.e., administrators).
	<i>Manufacturer</i>	Need not explicitly provide software tools for configuration of the network devices. By designing GANA conformant devices, interoperability with other GANA conformant devices is guaranteed.
	<i>End User</i>	Obtains the best possible performance level, no PHY layer configuration necessary.

The detailed Technical Description of the Scenario

In this Scenario, we consider installing a new network in an environment with other 802.11 networks under operation. We seek to observe and evaluate proper selection of radio channel(s) by a node auto-configuring in the new network. We also intend to observe that a change of the network configuration (due to the movement of the nodes, change of the transmission parameters) can trigger a channel(s) reallocation. The proposed scenario uses WARF extension to IPv6. WARF introduces four following new messages, which are implemented by using the Hop-by-Hop extension header option (see detailed description in Deliverable D-2.3 [3]):

- Channel Quality Report (CHQREPORT)
- Number_of_Interfaces_Report (INTFREPORT)
- Set Signalling Channel (SETSIGCH)
- Set Channel (SETCH)

To get advantage of the proposed extension, nodes must be WARF - conformant. Non-WARF-conformant devices ignore messages sent; because of extension headers type used (*skip over if unknown / may change en-route*, having the leading bits of 001). In the case when messages are ignored, a default channel allocation happens, which may lead to the suboptimal performance.

The first part of the proposed activity is built of two report messages, sent by the nodes either as separate packets or piggybacked to the user data:

- 1a. The CHQREPORT message: in this message all used channels are described using relevant parameters: Channel ID, SNR and Load.
- 1b. The INTFREPORT message: in this message all interfaces are described using Interface ID, Channel ID and TX power.

Goal of these messages is to provide the channel status information for the DEs. DEs collect the information and trigger the second part of the activity, which is built of two control messages:

- 2a. SETSIGCH - used to set a signalling channel and its TX power.
- 2b. SETCH - used to set user data channels and their TX power.

Channel selection can happen at the initialization stage and every time when collected information changes, so that the selection algorithm calculates a change in the channel configuration. Such a change can happen e.g. when one or more of the nodes involved in or affecting the communication process changes transmission parameters. A change can be initiated by different nodes and can be a result of different actions:

Node Types:

- A node itself,
- Node's direct neighbour,
- Node's indirect neighbour,
- Rogue devices and other sources.

Action Types:

- Movement,
- TX power change. There are also special cases: switch on (power change from 0), switch off (power change to 0), and channel change (sum of two power changes in two channels).
- Choosing a channel with the better (ideally, the best) SNR increases performance in two ways:
 - By decreasing (bit-, frame- and packet-) error rates, resulting in less frequent retransmissions and lower channel load,
 - By allowing for the more sophisticated, faster, but usually more error-prone modulation schemes; in 802.11 networks bitrates can vary from 1 to 54 Mbps.

The allocation algorithm takes into account status information from the neighbouring nodes. WARF provides only a communication framework (the format of the exchanged), while channel selection algorithms is not specified (intentionally). An overview of the existing algorithms for channel allocation have shown that a number of channel allocation algorithms can be implemented using these messages which consists of description of several channel allocation algorithms). The user may define its own algorithm for channel allocation. The choice of the algorithm for demo implementation will be done later and network operations will be compared with the classical approach, i.e. static channel assignment by the network operator

Summary of the key Features-Combinations that enable the realization of the Scenario

Scenario Name	Auto-Configuration of radio channels in 802.11 networks	
Protocol	Summary of the Features that enable the realization of this Scenario	Decision Element(s) involved and the Self-* functionality(ies)
IPv6 core protocols	Extension Headers	N/A
IPv6++	<ul style="list-style-type: none"> - New type of the Hop-by-Hop Option Extension Header (WARF Extension Headers) - CHQREPORT - INTFREPORT - SETSIGCH - SETCH 	<ul style="list-style-type: none"> - Resource monitoring DE, - Resource control DE.
Other types of enabling Functional Features	A proper algorithm must be used to perform the allocation process	N/A

A-1.1.3 Autonomic Routing Scenarios

A-1.1.3.1 Autonomic multipath routing in 802.11 Mesh Networks

Autonomic routing in mesh networks means that all the routing procedures are with a very limited supervision (no human intervention, intrinsic adaptive mechanisms) and they are able to adapt to changes related to available routing resources and actual usage of those resources. In practice, it means that the nodes must implement mechanisms for self-discovery of new networking resources, self-configuration/re-configuration according to the discovered resources, resource optimal usage and self-healing mechanisms that are able to cope with networking resource failures. All those mechanisms have to be combined together to provide fast, real-time actions in response to events. The approach will focus on loose-source multi-path autonomic routing in 802.11 mesh networks. The use of 802.11 in this use case provides many benefits in the highly dynamic networking environment. The presented approach will show efficient routing operations in such highly dynamic networking environments. The simultaneous adaptive multi-path approach will contribute to the optimization of networking resource usage (limiting congestion probability), and increase reliability in case of topology changes or link/node failures. All such operations will be autonomic at the node, sub-network or network level. For more details on all the issues, refer to D-1.3 [9] section 4.1.2.

The scenario for multi-path routing in 802.11 mesh network is aimed at presenting selected features of WARF, with the main emphasis put on multi-path auto-configuration. The Scenario considers one-domain ad hoc network environment. The scenario aims to present improvement of communication reliability and end-to-end throughput.

High-Level Description of the Scenario

Autonomic multi-path routing in 802.11 mesh network	
The Story-line	<p>Several nodes, having autonomic routing features of WARF platform implemented, try to form a one-domain IPv6 - based ad hoc network. The network is to be self-configuring, without the necessity of user's control, however user interactions are allowed to constrain the network behaviour. There is no need to have any operator or administrator to enable the network. Proposed solutions are beneficial to the end users, which do not have to manually configure their devices and can profit better throughput of the network.</p> <p>Once configured, the network is able to keep multi-path routing regardless of changes in the networking environment. Specifically auto-configuration procedures give support to situations where:</p> <ul style="list-style-type: none"> • New node is joining the network. • Cooperating node leaves the network. • Node failure temporally emerges at one randomly selected node. • Unexpected node disconnection. <p>The ability to maintain multi-path routing within a network, to reach the state where all nodes are properly configured in terms of multipath selection, is perceived as a measure of the autonomic routing capabilities and the level of resilience.</p>

Short description of the scenario	<p>Nodes are physically connected by means of wireless 802.11g capable interfaces and form an ad hoc network. They have the FreeBSD operating system with e.g., IPv6 KAME implementation in the system kernel, modified to enable multi-path routing. All nodes are aimed to form one domain connected with open Internet by two devices, which have IPv6 tunnels set to SixXS.</p> <p>The nodes exchange between themselves high volume data (audio-video streams) in a burst way. Occasionally they exchange emails with servers in the open Internet.</p>
Scenario Scenes	<p>Scenario is divided into several scenes:</p> <ul style="list-style-type: none"> • Node joining: auto-configuration of routing tables in the case where single new node wants to join the multi-path domain or when a node after a disconnection rejoins the network. • Node leaving: situation emerging when a node gracefully announces its will to leave the network. In this case, the network nodes recalculate multipath routes. • Temporal failure: A randomly selected node fails for a short time period (e.g., a moving object hides the node antenna). When the failure passes, the network is able to recreate the previous configuration quickly. • Unexpected disconnection: A randomly selected node disconnects without notification and in unexpected moment. The network is able to recognize the fact and to rebuild multi-path routing table respectively.
Current problems/limitations with <i>current practices</i> and/or <i>current technology</i>	In current MANET deployments there are no multi-path routing. Typically only one path is used for data transfer over the network in practice. It results with limited throughput of the communication. Moreover transfer interruptions occur in case of a node disconnection.
Network Environment	Wireless one-domain multi-path routing ad hoc network
<i>Autonomic multi-path routing</i> – What it solves and the benefits	Autonomic multi-path routing of WARF platform uses IPv6 due to its ability to distinguish individual data flows. If a data flow needs to go via a single path (e.g., a TCP connection), then the node level routing decision element is able to satisfy this need. Simultaneously other such a flow, between the same endpoints, can be directed via another path to higher up the network overall throughput. If a data flow doesn't need to go via a single path (e.g., an RTP connection), then the node level routing decision element can split packets of the flow between different paths to speed up the flow.
System(s) Involved	Every node having capability to act as a host and as a router

	Actor/Player	Benefits (with Rationale)
Key players that benefit	<i>Operator</i>	<p>The wireless one-domain multi-path routing ad hoc network is self-governing. Even though users are able to tune up some features (e.g., access rights), the multi-path routing is self-configuring. An interaction between the domain and an operator happens only on operator's access points.</p> <p>The multi-path feature can result in higher demands on data rate at the access points, and consequently result in higher income for the access.</p>
	<i>Manufacturer</i>	<p>Devices with the new technology implemented will allow using them for new applications for ad hoc networks (e.g., for tourist groups, scouts, pilgrims) and for existing applications but more efficiently. Thus this new feature will be interesting for a potential customer and would make the device supporting the technology much more competitive. A manufacturer that will offer such devices will have a chance to higher the sell volume.</p>
	<i>End User</i>	<p>End users will gain the possibility to use new applications, and will be able to enjoy more efficient ad hoc networks.</p>

The detailed Technical Description of the Scenario

The scenario aims to present autonomic behaviors such as auto-configuration and operation of AOMDV routing protocol in the case: there are no Network-Level Routing-Management Decision-Element, which can impose another routing protocol and its configuration.

An overview of WARF platform is given in D-2.1 [10], as well as an essential description of AOMDV. AOMDV is based on Ad hoc On-demand Distance Vector (AODV) Routing, RFC 3561. Below we give only description of interactions between the node decision elements, which are involved in the proposed scenario.

The WARF platform is an integral part of the GANA architecture. It is able to work in a separated domain and to interact with other domains. It can work without any network level controlling node. However if such a node appears, then it can influence the WARF behaviour. The network level controlling node can cooperate with ONIX in order to satisfy local network polices of the WARF domain neighbourhood. Figure 25 represents GANA architecture for WARF platform. The figure does not reflect the physical structure of the network. It shows only GANA elements and communication links between them.

In GANA architecture for WARF platform each node is equipped with Node Level Main DE, Node Level Routing Management DE and Node Level Auto-Configuration DE interacting with each other. Auto-configuration DE steers the auto-configuration process and manages several entities, like ND++_DE, responsible for stateless and stateful address auto-configuration. It is responsible for quick and efficient

IPv6 address configuration. The AC_DE conceptualization is out of the WARF scope. This DE provides needed addresses to the RM and Main DEs.

The Routing Management DE controls routing protocols by means of their corresponding configuration plug-ins. In this case the routing protocol is AOMDV and the configuration plug-in is AOMDV DE. RM DE reports to Main DE about the default routing protocol configuration. The Main DE is able to pass this information to the Network Level Controlling Node. The Controlling Node can request all Autonomic Nodes, in the domain, to switch routing protocol or to change its parameters. Main DE is responsible for communication with Network Level Controlling Node.

In the scenario we assume neither interaction with ONIX nor with Network Level Controlling Node. All nodes work autonomously. RM DE discovers all installed routing protocols (here we have only AOMDV) and decides which one to activate. Next it takes from AC DE addresses of the node and discovered neighbours, and passes them to AOMDV DE.

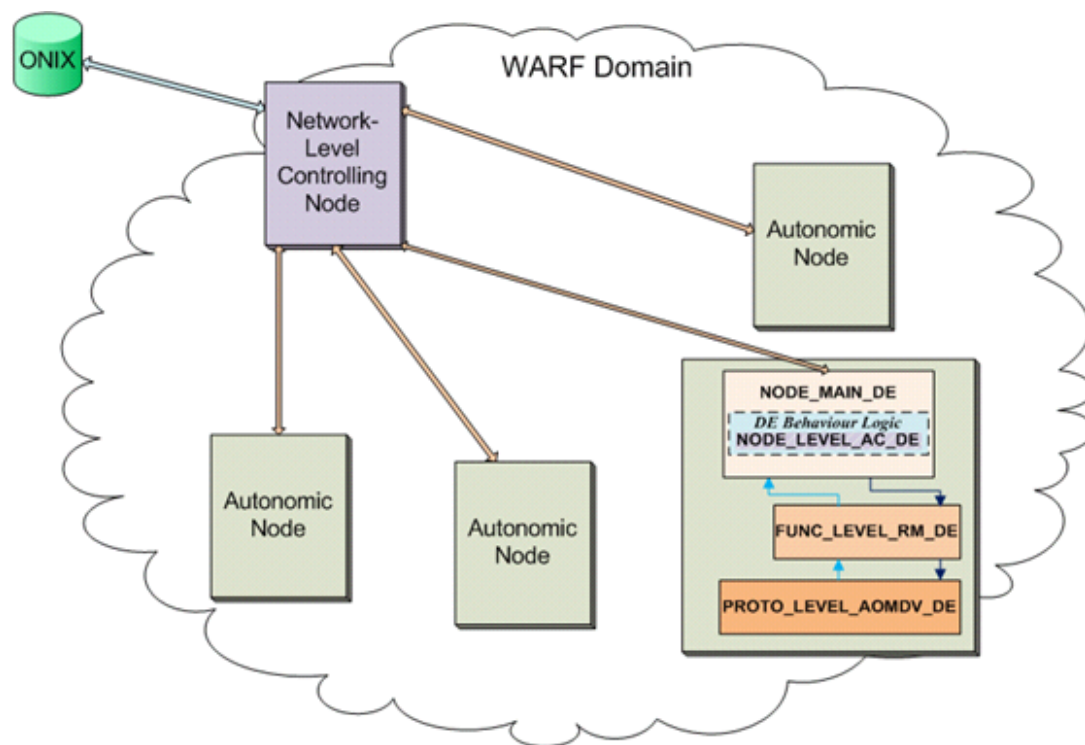


Figure 25: GANA architecture overview for WARF platform

After that it enables the routing protocol and takes from AOMDV DE the list of routing tuneable parameters and their default values. Subsequently it informs Main DE about selected routing protocol and its configuration. Main DE can report this information outside the node for diagnostic purposes or to a network level controlling node.

Periodically RM DE takes routing statistics from AOMDV DE and analyses them. It can decide to change some tuneable parameters passing to AOMDV DE their requested values. The AOMDV protocol has full responsibility for routing. The described in previous chapter scenes are supported by AOMDV protocol instances in concerned autonomic nodes.

To observe the network behaviour, the nodes will generate some data packet traffic that simulates video streams inside the domain and email communication with servers outside the domain. An observation node will be installed in the domain to periodically query Main DE of all the autonomic nodes about routing configuration and statistics.

Summary of the key Features-Combinations that enable the realization of the Scenario

Scenario Name	Autonomic multi-path routing in 802.11 mesh network	
Protocol	Summary of the Features that enable the realization of this Scenario	Decision Element(s) involved and the Self-* functionality(ies)
IPv6 core protocols	<ul style="list-style-type: none"> • IPv6 transports AOMDV messages • IPv6 flow label allows for path selection when it is needed 	<ul style="list-style-type: none"> • NODE_MAIN_DE • FUNC_LEVEL_RM_DE
IPv6++	<ul style="list-style-type: none"> • It is reactive protocol • Created paths are disjoint in terms of nodes • Route maintenance process is combined with time-outs and HELLO packets 	<ul style="list-style-type: none"> • NODE_MAIN_DE • FUNC_LEVEL_RM_DE
Other types of enabling Functional Features	N/A	N/A

Autonomic routing in mesh networks means that all the routing procedures are with a very limited supervision (no human intervention, intrinsic adaptive mechanisms) and they are able to adapt to changes related to available routing resources and actual usage of those resources. The presented approach focuses on loose-source multi-path autonomic routing in 802.11 ad hoc networks. The use of 802.11 in this use case provides many benefits in the highly dynamic networking environment. The routing operations are efficient in such highly dynamic environments. The simultaneous adaptive multi-path approach contributes to the optimization of networking resource usage (limiting congestion probability), and increases reliability in case of topology changes or link/node failures. All such operations are autonomic at the node level.

A-1.1.3.2 Autonomic Routing and Self-Adaptation driven by Risk-Level Assessment in Fixed Network Environments

In this Scenario, we seek to demonstrate the benefits brought by a *Self-Adaptation* behaviour implemented by the **Routing-Management-DEs** based on *tuning OSPFv3 link weights in order to dynamically adapt to Risk-levels in the network elements and the network*—thereby causing traffic flow (or critical traffic) to avoid nodes with certain risk-levels. The Risk-Level information is communicated to the DEs by the *Risk Assessment functions* of the nodes/network.

During the lifetime of a network element and the network as a whole, failures often occur and should be avoided from occurring in the first place and so, there must be some mechanisms inbuilt into the node/network architectures of a self-managing network that assess the risk-levels associated with the usage of individual functional entities of a node/network at any time during their operation time in order to proactively avoid failures, damages, eventual outages and service unavailability.

In every network the risk of failure of a network element or a given area changes with time, depending on multiple factors such as the traffic volume and types, the age of network elements as well as external conditions. Therefore, the *Risk Assessment functions* implemented in individual GANA conformant nodes as well as on the network level must take into account the fluctuating probability of failure occurrence.

The failure prediction is based on the monitoring of internal state and behaviour of a node involving specific parameters, which are significant enough to predict a future failure, such as the hardware temperature, the internal fan status, the hard drive status, the voltage instability, the signal quality. Also, software related parameters must be taken into account, including the load, anomalies (memory leaking, unreleased file locks, file descriptor leaking, and data corruption), virus and hacker attacks, and the maintenance activities since such factors account for 20% of all failures. The failure prediction becomes richer by using machine learning methods, based on Support Vector Machines or Semi-Markov Processes, and thereby self-learn new node states, patterns or behaviors forewarning of failures in order to autonomously feed and refine the predicting-failure database.

High- Level Description of the Scenario

<i>Self-Adaptation of Routing as driven by Risk-Level Assessment in a Fixed Network Environment</i>	
The Story-line	Today's network technologies, network management systems and associated network management practices constrain the network operator to exercise Fault-management in a static manner without having the ability to take into account and address the dynamicity of the risk endangering their networks. That why a <i>Risk-Assessment functions</i> of nodes/network should provide real-time risk level assessment information in order to proactively prompt the nodes and the network to reactive as the probability of failure evolves by adaptively employing strategies that proactively avoid failures, damages, eventual outages and service unavailability. The adaptation behaviors must be triggered in real-time in response to the incurred risk. One example of adaptation strategies is that of implementing adaptive traffic engineering mechanisms at the routing level that dynamically <i>tune OSPFv3 link weights in order to dynamically adapt to Risk-Levels in the network elements and the network</i> .

	<p>The risk assessment information includes the details relevant for Decision Elements (DEs) to co-operatively employ the adaptation strategies, some of which must be employed within the autonomic fault-management and proactive resilience mechanisms at different levels of the DEs hierarchy of the GANA network. Risk-Level Assessment Information incorporates the following items:</p> <ul style="list-style-type: none"> The origin of the prediction (like CPU overheating), The probability of occurrence, The potential impact of the eventual failure Its severity The failure extent and the impacted services. <p>The manufacturer of the GANA conformant routers guarantees the operator that the Routing-Management-DEs embedded within the routers adaptively collaborate with the Network-Level Routing-Management-DE in order to proactively avoid failures, damages, eventual outages and service unavailability in response to the Risk-Levels communicated to them by the <i>Risk Assessment functions</i> of the nodes/network. The adaptation is done by <i>tuning OSPFv3 link weights in order to dynamically adapt to Risk-Levels in the network elements and the network</i>—thereby causing traffic flow (or critical traffic) to avoid nodes with certain risk-levels.</p>
Short description of the scenario	<p>The Scenario demonstrates <i>Self-Adaptation of Routing</i> as driven by Risk-Level Assessment in a Fixed Network Environment. The Routing-Management-DEs within routing nodes collaborate with the Network-Level Routing-Management-DE in <i>tuning OSPFv3 link weights in order to dynamically adapt to Risk-Levels in the network elements and the network</i>—thereby causing traffic flow (or critical traffic) to avoid nodes with certain risk-levels.</p>
Current problems with current practices and/or current technology	<p>In current IP networks, the routing function doesn't take into account the network elements risk of failure, which is a problem when a failure occurs. The routing only depends on traffic engineering metrics without taking into account the risk level of the network elements. Failures are reactively handled by the fault-management protocol thanks to the Hello protocol that periodically sends hello messages in order to detect appeared failures. Such technique may cause loss of packets, temporary routing loops, asymmetrical routing, or routing "black holes" because of delay between the failure occurrence and the OSPF convergence, The risk a failure needs to be taking into account in the routing protocol.</p>
Network Environment	Fixed/Wired

Self* Functionalities introduced	Self- *Functionality	Problems/Limitations it addresses
	Self-Adaptation	Current routing protocols don't take into account forewarning of failure information but only rely on reactive fault-management system to retrieve from a failure. Even if a lot of accurate reactive approaches exist, a proactive approach that would anticipate the failures will be able to minimize the damage of the menacing failures. OSPFv3 fault-management mechanism is triggered by a reactive failure detection that can be improved by proactive mechanisms which would use risk-level information to avoid risked network elements.
Self-Adaptation – What it solves and the benefits	The Self-Adaptation feature aims to anticipate failures by using risk-level information for the routing protocol. The process use link weights to prompt the traffic to avoid nodes with certain risk. This mechanism allows bypassing the fault-management slowness and avoiding visible impact of failure on the traffic flow. The failure anticipation benefits recovery speed, since the preventive failure bypass enable to isolate failure before it cause damages.	
System(s) Involved	<ol style="list-style-type: none"> 1. Core Routers 2. Edge Routers 3. Access Routers <p><i>Note: Each of these devices must have been designed to conform to GANA and therefore must embed, apart from the NODE_MAIN_DE, the Function-Level Routing-Management Decision-Element (FUNC_LEVEL_RM_DE).</i></p>	
Key players that benefit	Player	Benefits
	<i>Operator</i>	The Operator will easily respect its SLA requirements, thanks to improved services availability.
	<i>Manufacturer</i>	The manufacturers will provide a feature that will help their customers interested in improving the reliability of theirs networks.
	<i>End User</i>	The better reliability will improve the quality of experience of the end-users who will not be affected by the failures anymore.

The detailed Technical Description of the Scenario

The scenario for the demonstration is divided into 3 scenes, each showcasing the various features of the GANA elements with respect to the risk-aware self-adaptation functionality embedded in them. However the risk-aware self-adaptation functionality described here is confined to the Autonomic Routing domain. **Scene 1** (see next) describes how a node does local risk evaluation and takes corresponding risk mitigation actions based on this. In **Scene 2** (see next), we describe how risk mitigation actions based on both local risk assessment and global risk assessment is carried out. Finally, in **Scene 3** (see next), we describe how the global risk assessment and mitigation solutions are produced when local risk assessment is impossible or a local risk mitigation solution is unachievable.

Assumptions

Initially, all the nodes and devices in the network are assumed to have performed the Auto-Configuration and Auto-Discovery processes described in previous sections and scenarios. The network is now fully functional with the routers performing their routine tasks. Further, it is seen that during the Auto-Configuration process when the network gets partitioned into OSPF areas, both CR1 and BR1 are available as viable border router candidates based on their capabilities. For some reason (as decided by NET_LEVEL_RM_DE), BR1 is chosen to play the role of a border router, while CR1 is assigned the role of a core router. The link between CR1 and BR5 is configured. However, CR1 does not have an OSPF interface with BR5 on this link (i.e. there is not exchanging OSPF messages), though it is configured to forward data. Alternatively, it is possible that CR1 listens to LSAs sent on this link by BR5, but without sending its own LSA along this link. Bob (a user) is connected to BR5. Other users, Alice, Charlie and Dave are connected to BR4. The content providers, CP1 (CPIA) and CP2 are connected to BR3. CP2A is connected to BR2. Further, it is assumed that the SLA between the ISP and Content Provider (CP1, refer Figure 26), requires the ISP to give higher priority and bandwidth for the data/packets from CP1. The scenario of content is assumed to be running before the start of this scenario such that a movie or video from CP1 is assumed to be streamed un-interrupted to Bob.

A Risk Model is defined as a meta-model (information model) that contains items (but not limited to) such as;

- The origin of the prediction (like CPU overheating),
- The probability of occurrence,
- The potential impact of the eventual failure
- Its severity
- The failure extent and the impacted services.

The Risk Model is based on known behaviors and patterns forewarning of failure. Such Model relies on Bayesian networks to represent the probabilities of failure, their relations, and the role of the monitored parameters forewarning of failure, in order to disseminate consistent risk information.

The **Risk Model** that is used by NODE_LEVEL_R&S_DE and NET_LEVEL_R&S_DE for the computation and correlation of the local **Risks Description** and the **Global Risks Description** respectively is assumed to be built-in to the DEs at design time. Based on this risk model, NODE_LEVEL_R&S_DE calculates the risks for a node with respect to the information it collects from or is provided by the monitoring components and from the underlying DEs. Once the Risks Description of a node is computed, they are disseminated at run-time to the underlying DEs and NODE_LEVEL_FM_DE for the required risk mitigating actions. If NODE_MAIN_DE decides that the Risks Description computed by NODE_LEVEL_R&S_DE have crossed a certain risk threshold or

severity, it will disseminate this Risks Description to ONIX. NET_LEVEL_R&S_DE is assumed to be subscribed to receive Risks Description from ONIX after its Auto-Configuration process. The other Network-Level-DEs are assumed to be also subscribed (either through normal subscription or through an on-behalf subscription) to receive either a part or the complete aggregated and correlated Global Risks Description.

Scenarios Description

Initially, all routers in Area 1 and Area 2 are assumed to be functioning without any errors with their hardware temperatures well below the safety threshold limit.

Scenario 1: Local Risk Evaluation and Local Risk Mitigation Actions

Scene 1 – Temperature of BR1 reaches threshold limit – Moderate Risk of Router failure

As the scenario progresses, after some time all the routers are functioning properly, with the exception of BR1 whose hardware temperature has just reached its threshold limit. This is indicated by the thermometers shown next to the routers in Figure 26. As it can be seen from the figure all the routers' thermometers with the exception of BR1 show low temperature readings. NODE_LEVEL_R&S_DE of each router based on the information supplied by monitoring components of the node and from the sensor and non-sensory interfaces of the underlying DEs and MEs computes the Risks Description of the node. Based on the severity of the Risks Description, NODE_MAIN_DE will decide to escalate and disseminate the Risks Description to ONIX for further risk analysis by NET_LEVEL_R&S_DE. At the moment, NET_LEVEL_R&S_DE considers this risk to be persistent and not transient even though its current severity is only moderate. At this stage, NODE_MAIN_DE of BR1 computes the need to disseminate the Risks Description to ONIX as the hardware temperature of the router reached the safety threshold limit. However, in this case, this dissemination of the Risks Description by NODE_MAIN_DE of BR1 is carried out for cautionary notification purposes only. The Function-Level-DEs of the node are able to handle the current severity of Risks Description locally without a need for a solution from the Network-Level-DEs. In addition to the Risks Description from BR1, two more routers, CR1 and BR2, disseminate their Risks Description to ONIX (message 1). It is assumed in this scene that the Risks Description disseminated by NODE_MAIN_DE of BR1 include the Risks Description due to the rise in the hardware temperature of the router beyond the threshold limit. The Risks Description disseminated by CR1 and BR2 may or may not be related to the risks of BR1.

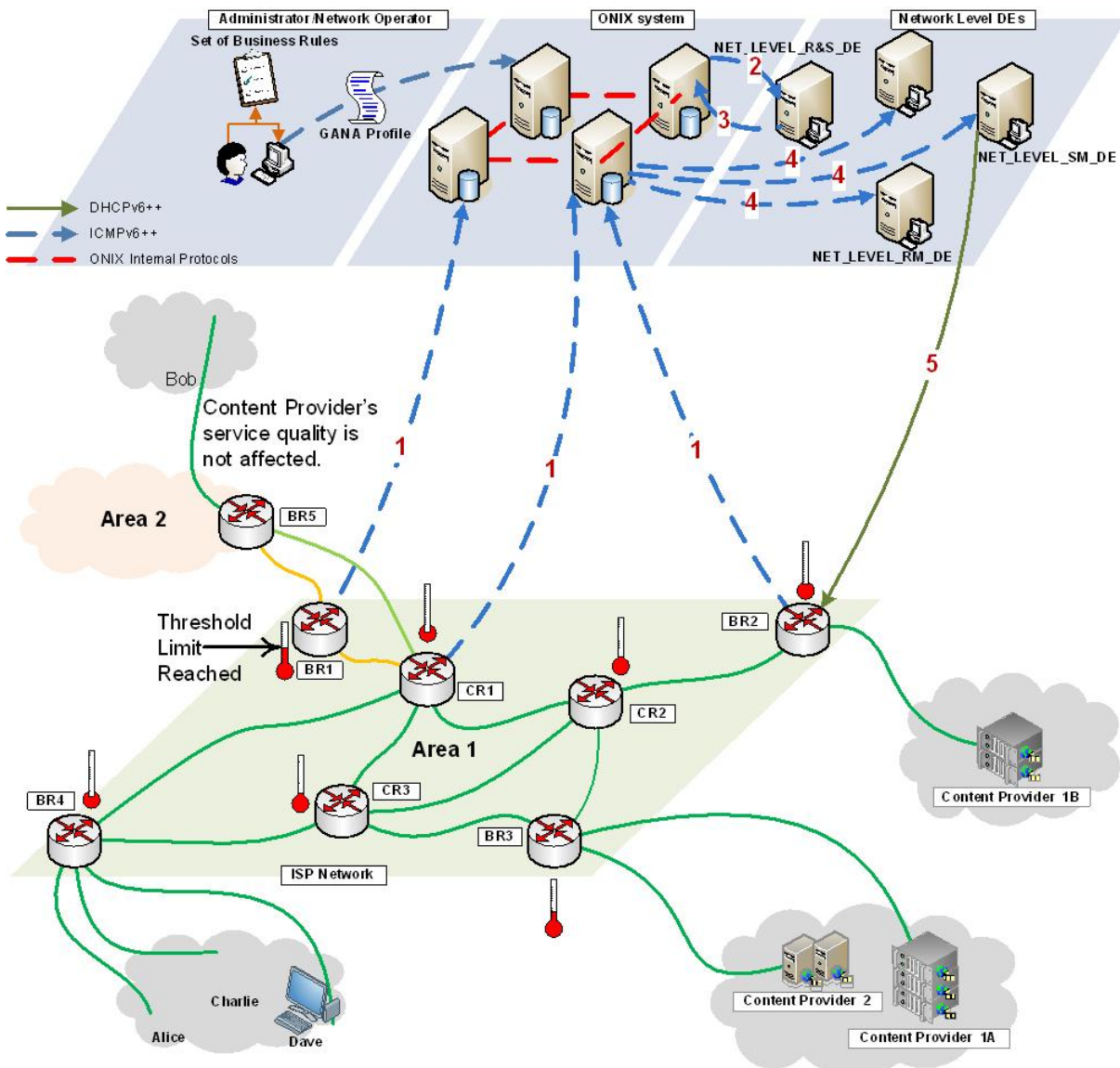


Figure 26: Temperature of BR1 reaches threshold limit – Moderate risk

When NODE_MAIN_DE of BR1 disseminates Risks Description to ONIX, it simultaneously disseminates the same to all the underlying DE's and NODE_LEVEL_FM_DE. Based on the Risks Description obtained from NODE_MAIN_DE, FUNC_LEVEL_RM_DE of BR1 computes the cost of the link connecting BR1 to CR1. The new cost of this link based on the node's Risks Description is computed by the "Risk Aware Link Cost Computation" block of FUNC_LEVEL_RM_DE. This is shown by messages 1 and 2a of the magnified GANA node in Figure 26. In the mean time ONIX pushes the Risks Description of BR1, CR1 and BR2 to the NET_LEVEL_R&S_DE. NET_LEVEL_R&S_DE aggregates and correlates the Risks Description and computes the Global Risks Description for the network, and publishes it to ONIX (message 3). ONIX pushes a part or the complete Global Risks Description (based on the type of subscription that was made with the ONIX) to the other Network-Level-DEs (message 4). In the scene, NET_LEVEL_RM_DE does not provide any solution to BR1 as it is aware of the severity of the risks of BR1 and its implications. However, it is possible that NET_LEVEL_RM_DE provides a solution for risk mitigation, though not in this scenario. Nevertheless, NET_LEVEL_RM_DE registers

the occurrence of such a risk (to be used in future computations) in the network. Other Network-Level-DEs such as NET_LEVEL_SM_DE can provide mitigation actions for security related risks, such as for BR2 (message 5).

Once a new link cost (moderately higher than the previous link cost) is computed as risk mitigation action, FUNC_LEVEL_RM_DE changes the link cost between BR1 and CR1. This is indicated by the orange colour of the link between BR1 and CR1. LSAs indicating the new link cost are advertised through the network and routing tables are updated based on this. This new link cost prohibits (discourages) the forwarding of a large volume of traffic along that link, and forces the other routers to forward some data along other paths, minimizing the traffic flow through BR1. The link weights of the other links remain unchanged. This behaviour is kept until the current risk is mitigated, and till the link cost has been restored to a lower or older value. By minimizing the traffic through BR1, the risk of packets being lost due to router failure is minimized. The node thus self-adapts its routing behaviour and consequently it's forwarding behaviour (and influences the routing and forwarding behaviour of its neighbours) with respect to the local risks evaluation. The service (HD content streaming) from CP1 to Bob remains unaffected throughout this scene.

Scenario 2: Global and Local Risk Evaluation, and Local and Global Risk Mitigation Actions

Scene 2 – Temperature of BR1 crosses threshold limit – High Risk of Router failure

Even though the risk of packet loss is minimized by the local risk mitigation actions of the router BR1, the hardware temperature increases further. This indicates the presence of external problems, such as router fan or cooling failure rather than router operation per se. As the hardware temperature of BR1 goes beyond the threshold limit, NODE_LEVEL_R&S_DE computes the new Risks Description of the node and disseminates it simultaneously to both the ONIX and the underlying DEs and NODE_LEVEL_FM_DE. Based on the local risk evaluation, FUNC_LEVEL_RM_DE's "Risk Aware Link Cost Computation" block performs mitigation actions by computing new link cost and applying it to the link between BR1 and CR1 (message 2a, and say darker orange to the link).

At the same time, ONIX pushes the Risks Description to NET_LEVEL_R&S_DE (message 2b). The DE performs the same actions as described previously, and updates the Global Risks Description data in ONIX (message 3). ONIX pushes parts or the complete form of this updated risks data to the other Network-Level-DEs (message 4). In this scene, NET_LEVEL_RM_DE, based on previous knowledge and current data, computes a solution for risk mitigation and avoidance in BR1 and the network as a whole. The solution is pushed to BR1 (message 5) and the other routers in the network. NODE_MAIN_DE of BR1 and the other routers relays this solution to their FUNC_LEVEL_RM_DE which in turn applies the provided link cost (high for BR1-BR5 link and low for the other links) to the links between them (message 6, link color red for BR1-BR5 link, link color dashed-green for the other links between the core routers).

OSPF's LSAs with the new (high) link cost are advertised and the routers recalculate their routing tables based on this. This high link cost aggressively prohibits the use of router BR1. Only certain type of traffic (e.g. high priority traffic from CP1 to Bob) for the current risk level is allowed to be forwarded by BR1. All other traffic is re-routed. Thus both the Node and the Network-Level-DEs participate to adapt the routing behavior and consequently the forwarding behavior of the network based on the risk evaluation and analysis. The quality of the service provided by CP1 to Bob remains unaffected even though the operating conditions of BR1 deteriorate.

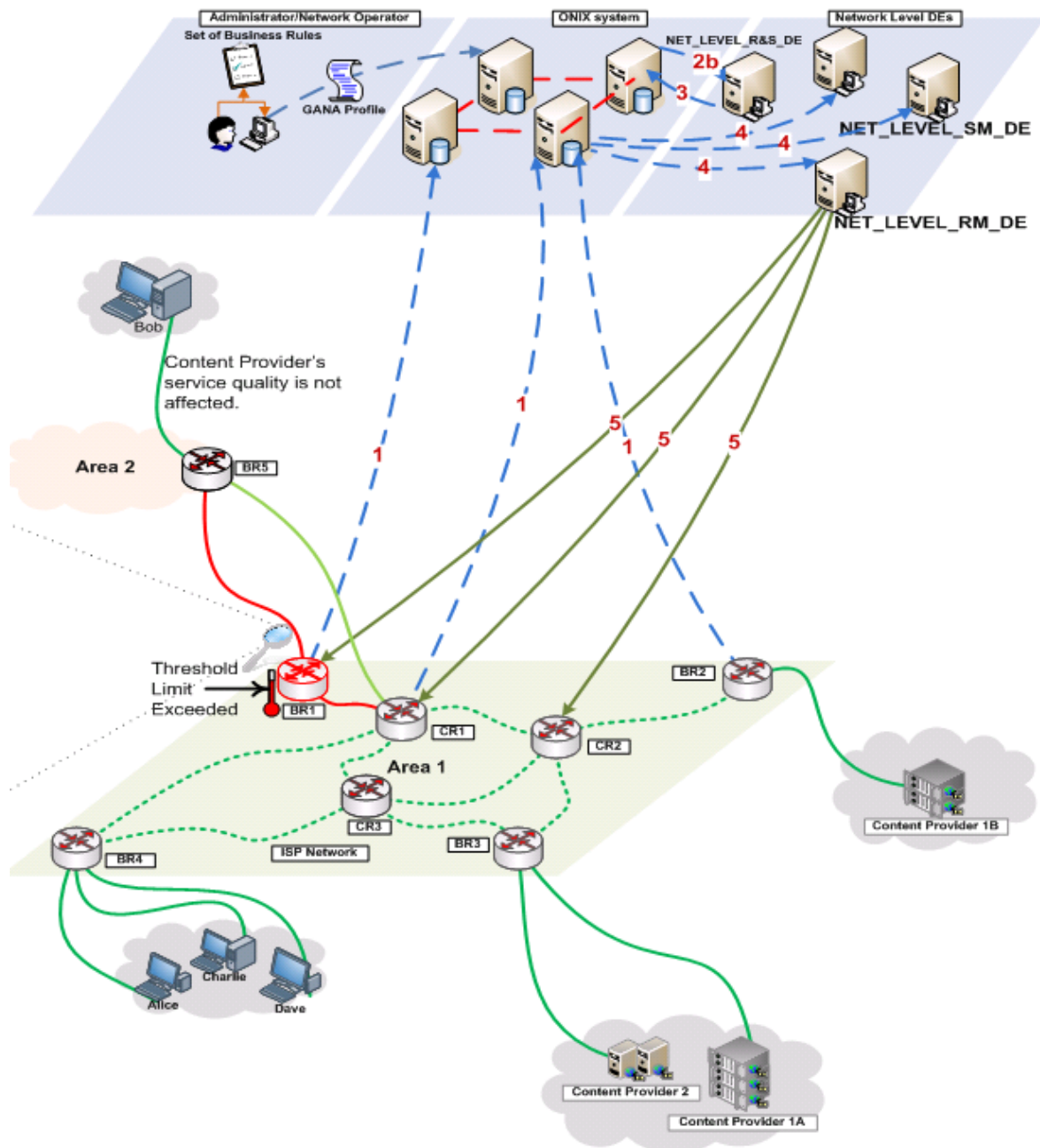


Figure 27: Temperature of BR1 crosses threshold limit – High Risk

Scenario 3: Global Risk Evaluation and Global Risk Mitigation Actions

Scene 3 – Temperature of BR1 reaches Maximum limit – Maximum Risk of Router failure

As the temperature of BR1 further increases to the maximum, as indicated by the thermometer in Figure 27, the above described messages sequence from 1 to 4 plays through the scenario. However, in this case, the node is unable to find a solution or a corresponding link weight that mitigates the risk. Based on the previous computation and current risk data, NET_LEVEL_RM_DE predicts that BR1 has a high probability to fail. Based on this computation, it instructs a role change for the routers in the network. NET_LEVEL_RM_DE issues four different messages, one to BR1 (message 5a), the other to CR1 (message 5b), the third to BR5 (message 5c) and the fourth to the core routers in Area 1 (message 5d). Through message 5a BR1 is instructed to shutdown to prevent further damage to the router. With message 5b a *Profile* is pushed to CR1 and is instructed to switch to the role of a border router (the role previously played by BR1). With message 5c, BR5 is instructed to re-configure its link interface with respect to the new role of CR1 (the new BR1), and finally with message 5d, the other core-routers are instructed to change their link weights to reflect the requirements of the network topology and to achieve the network goals. This is indicated by “dot & dashed-green” link connecting the core-routers, a thick green link connecting BR5 to CR1 (new BR1), and a red link with an **X** connecting BR1 (now XX) to BR5 and CR1 (new BR1). Thus the Network-Level-DEs adapt the routing behavior and consequently the forwarding behavior of the network when the local risk mitigation strategies of a node are unable to find a solution. The HD content streaming service to Bob remains unaffected during the role switch. This would be indicated an un-interrupted movie streaming at Bob’s display console (see Figure 28)

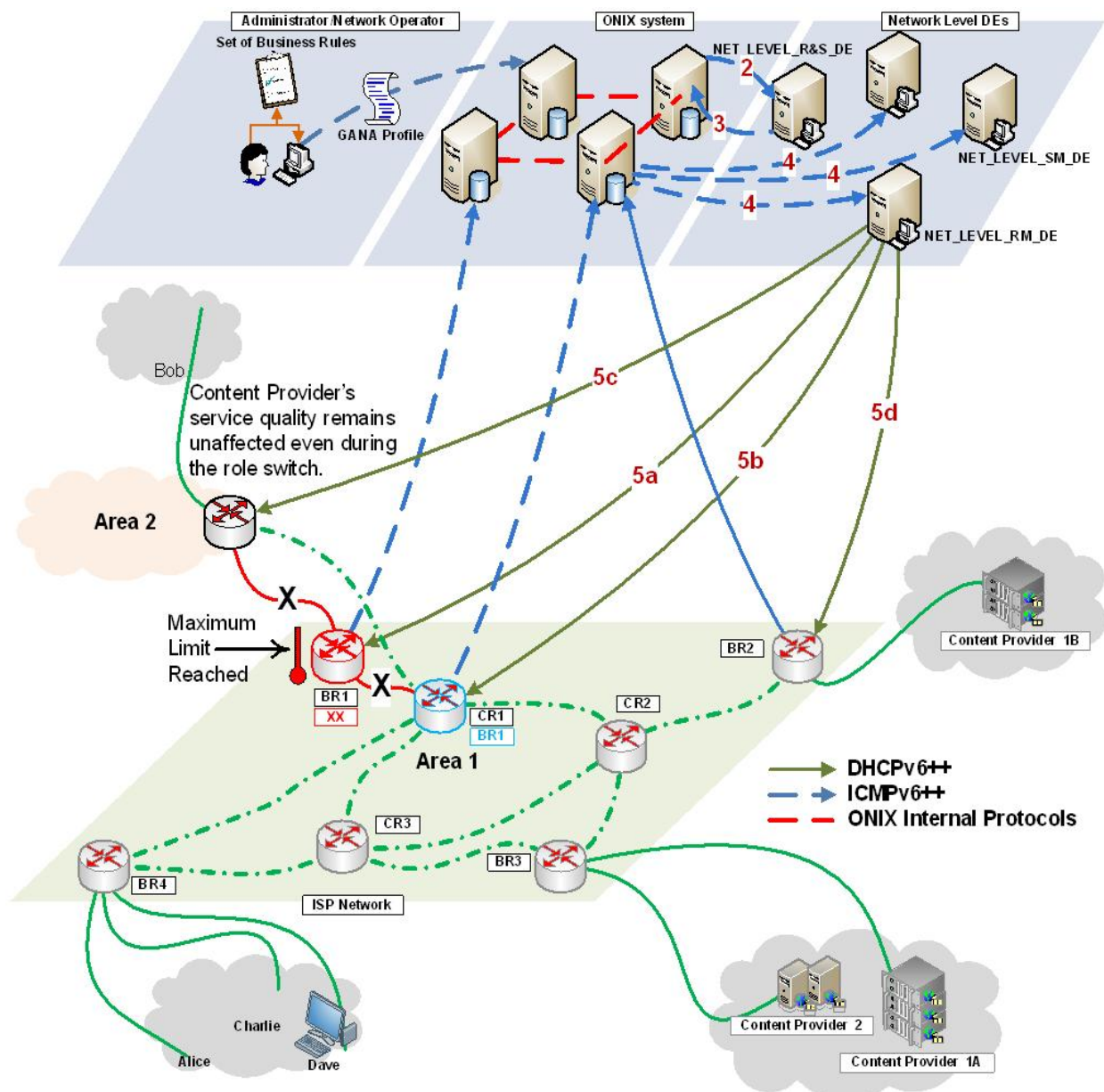


Figure 28: Temperature of BR1 reaches threshold limit – Maximum Risk

Summary of the key Features-Combinations that enable the realization of the Scenario

Scenario Name	<i>Self-Adaptation of Routing</i> as driven by Risk-Level Assessment in a Fixed Network Environment	
Protocol	Summary of the Features that enable the realization of this Scenario	Decision Element(s) involved and the Self-* functionality(ies)
IPv6 core protocols OSPFv3 (the OSPF version for IPv6)	Stateless address auto-configuration or the use of some more advanced address configuration methods such as the one proposed for the Autonomic DHCP architecture applied to a SOHO network	NODE_LEVEL_AC_DE (self-configuration)
	Adjustment of Link cost and recalculation of routing tables.	FUNC_LEVEL_RM_DE (Self-Adaptation using Risk Aware Link Cost Computation block)
DHCPv6++	Add/Update Risk Description to the ONIX. Retrieve/Subscribe Risk Description from the ONIX.	<ul style="list-style-type: none"> - NODE_MAIN_DE - NODE_LEVEL_R&S_DE (Self-Healing) - NET_LEVEL_RM_DE (Self-Adaptation) - NET_LEVEL_R&S_DE (Self-Healing)
ICMPv6++	Exchange of Risk Mitigation actions between the DEs. Acknowledgement of the mitigation of Risk.	<ul style="list-style-type: none"> - NODE_MAIN_DE - NET_LEVEL_RM_DE - NET_LEVEL_R&S_DE
Other types of enabling Functional Features	<p>In this scenario the emphasis is on Self-Adaptation based on the risk description of a node and global risk description of the network and, how it can lead to more complex self-* functionality (self-optimization).</p> <p>We assume that monitoring would play a role in the collection of data for the generation of Risk Description inside a node.</p>	N/A

A-1.2 Advanced Autonomic Networking Scenarios (WP3)

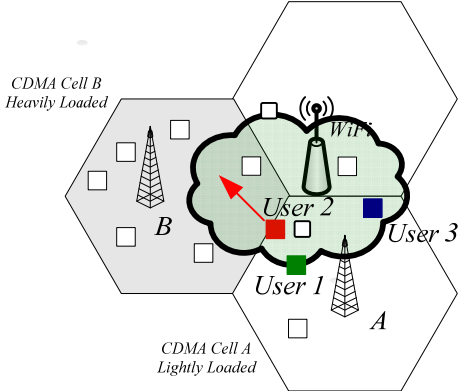
A-1.2.1. Autonomic Mobility and QoS Management Scenario

The objective of this scenario is to highlight and demonstrate the functionalities of the autonomic mobility and QoS management architecture for an integrated heterogeneous wireless environment, as they have been proposed and developed within the framework of project EFIPSANS (WP3). In this scenario the emphasis is placed in the demonstration of the self-adaptation and self-optimization functionalities of an autonomic mobile node, which are enabled by the proposed architecture, and therefore reveal the potential emerging benefits from users' and networks' perspective.

High- Level Description of the Scenario

Autonomic Mobility and QoS Management over an Integrated Heterogeneous Wireless Environment	
The Story-line	<p>The proposed scenario is taking place on a crowded square of London. Three unknown to each other people (George, Maria and John) are requesting access to the wireless network for deferent purposes. The autonomicity of their mobile devises triggers a sequence of events towards self-optimizing the requested users' service performance, which will eventually affect each other's behavior.</p> <p>George works in a prestigious IT company and is heading towards a meeting with important partners. He is not aware of the meeting place as well as the agenda. Therefore, he uses his mobile device (e.g. PDA) in order to download the required information from his email account. George is unaware of the available access networks in his locality. His autonomic mobile terminal assess the status of the available networks, in terms of current load, QoS-provisioning capabilities as well as George's service QoS prerequisites and selects the most appropriate network to be attached to. Thus, its corresponding DEs make all the appropriate actions towards enabling the requested service. As a result George is using the WiFi network.</p> <p>At the same time Maria is walking in an area close to George's location and is already speaking on her mobile phone. Maria is walking towards the boundaries of the CDMA cell (cell A) that her mobile node is currently attached to and therefore, a handoff is initiated. In our scenario the targeted CDMA cell (cell B) is overloaded; thus, if Maria's cell phone will be attached to CDMA cell B, her call will be dropped. In our case, due to her mobile phone's autonomicity, it senses the status of the CDMA cell as well as the existence of the WiFi network and hence, selects the second one as the new point of attachment. Moreover, Maria's phone self-adaptation and self-optimization mechanisms prevent the degradation of her service in a proficient and seamless way.</p> <p>However, as time evolves Maria's mobile phone attachment to the WiFi network caused congestion. Thus, Maria's movement and autonomic mobile phone's actions, affect John, a nearby student sitting at a café, who is currently having a phone-conference via the WiFi network. His mobile device (e.g. PDA) senses that the WiFi network lacks of available resources and thus, the potential degradations of his user's (John) service. Therefore, initiates a</p>

	QoS-triggered vertical handoff (even if John is a static user) towards the available lightly loaded CDMA cell (cell A). This action not only reassures John's service high performance continuity, but also favors the overall network due to the reduction of the load of the WiFi access point.	
Short description of the scenario	This scenario illustrates the key novelties and functionalities of the autonomic mobility and QoS management architecture. Our goal is to demonstrate some basic self-* functionalities of an autonomic mobile mode which are enabled by the proposed architecture, as well as to reveal the emerging benefits from both users' and networks' perspective.	
Scenario Scenes	1 st Part: A new user is entering the system – QoS-aware autonomic network selection. 2 nd Part: A roaming users – Mobility-triggered QoS-aware vertical handoff. 3 rd Part: QoS-triggered vertical handoff.	
Current problems/limitations with current practices and/or current technology		
Network Environment	Heterogeneous Wireless Environment (CDMA, 802.11 WLAN)	
Self* Functionalities introduced	Self-*Functionality	Problems/Limitations it addresses
	Self – adaptation	<u>Current Practices:</u> Current practices concerning self-adaptation on wireless heterogeneous environments in terms of network selection in case of environmental, QoS or mobility triggering events are limited and mainly based in conventional myopic criteria such as signal strength or service differentiation.
	Self – optimization	<u>Current Practices:</u> The lack of self-adaptation capabilities in current networking framework described earlier results in limited practices towards nodes' or network's performance optimization.
Self – adaptation: What it solves and the benefits	Self – adaptation of autonomic nodes when roaming over a heterogeneous wireless environment can be realized by a) constantly monitoring various parameters of the changing environment b) processing, combining and assessing such information, c) reacting on specific triggering events by dynamically adapting the operation of some of the key autonomic functionalities and mechanisms (e.g.	

	mobility protocols, resource allocation mechanisms)	
Self – optimization: What it solves and the benefits	<ul style="list-style-type: none"> - Self-optimization of autonomic nodes, in terms of acting or reacting on QoS – triggering events towards optimizing their services’ performance, as well as fulfilling their corresponding QoS prerequisites. - Self-optimization of autonomic integrated networks, in terms of: a) increasing overall network’s performance (e.g. cells’ capacity, users’ QoS prerequisites fulfillment), and thus the number of the users that can be simultaneously and efficiently served; b) performing fast horizontal or vertical handoff, which enables seamlessly and efficiently mobility support. 	
System(s) Involved	 <ul style="list-style-type: none"> • Two CDMA base stations and their corresponding cells namely, cell A which is assumed to be lightly loaded and cell B which is assumed to be heavily loaded WLAN Access Point • A WiFi access point tightly coupled with the CDMA cells. • Three autonomic users (i.e User 1, requesting data, non-real-time services and User 2 and User 3 requesting real-time multimedia services). 	
Key players that benefit	Actor/Player	Benefits (with Rationale)
	Operator	Revenue maximization and optimal utilization of the network resources under given QoS constraints in both WLANs and CDMA cellular networks while achieving overall system’s load balancing.
	Manufacturer	None
	End User	Achieves services’ QoS optimization in terms of quality and performance while reducing its expenses.

The detailed Technical Description of the Scenario

During the evolution of the proposed scenario, the following DEs will communicate, interact and collaborate, synchronously or un- synchronously, either residing at cells' base stations (or access points), or at the autonomic nodes:

Mobility Related Functionalities:

NET_LEVEL_MOM_DE	Network-Level Mobility-Management Decision-Element
FUNC_LEVEL_MOB_M_DE	Function-Level Mobility-Management Decision-Element (at the mobile node)
PROTO_LEVEL_MIPv6_DE	Protocol-Level Mobile-IPv6 Decision-Element (at the mobile node)
PROTO_LEVEL_PMIPv6_DE	Protocol-Level Network-Based-Mobility-Management Decision-Element

QoS and Resource Allocation Related Functionalities:

NET_LEVEL_QoS_M_DE	Network-Level Quality-Of-Service-Management - (Wireless - Network) Decision-Element
FUNC_LEVEL_QoS_M_DE	Function-Level Quality-Of-Service Management (Wireless) Decision-Element (at the mobile node)
PROTO_LEVEL_NODE_R&Q_WLAN_DE	Protocol-Level Node's-Resource-Allocation-And-Quality-Of-Service-Management-In-WLAN-Networks Decision-Element
PROTO_LEVEL_AP_R&Q_WLAN_DE	Protocol-Level Access-Point-Resource-Allocation-And-Quality-Of-Service-Management-In-WLAN-Networks Decision-Element
PROTO_LEVEL_NODE_R&Q_CDMA_DE	Protocol-Level Node's-Resource-Allocation-And-Quality-Of-Service-Management-In-CDMA-Cellular-Networks Decision-Element
PROTO_LEVEL_BS_R&Q_CDMA_DE	Protocol-Level Base-Station-(eNode B)-Resource-Allocation-And-Quality-Of-Service-Management-In-CDMA-Cellular-Networks Decision-Element

The GANA Ontology oriented Analysis

In the following a detailed description of the proposed scenario is provided. The scenario includes three different parts (i.e. autonomic users' actions) that will take place sequentially in order to better demonstrate specific features and autonomic functionalities. Moreover, for each part we outline the sequence of actions that will take place, as well as the involved corresponding DEs.

Scenario's 1st Part: A new user is entering the system – QoS-aware autonomic network selection.

Action: An autonomic mobile node (User 1) is entering the area of an integrated heterogeneous wireless environment and is requesting a non-real-time data service.

The purpose of this part is twofold. On one hand, it aims at illustrating the key components/DEs of the architecture and the actions/decisions that the corresponding control loops are responsible for, while on the other hand it will demonstrate the multi-objective, multi-criteria decision making operation of an autonomic node's **FUNC_LEVEL_QoS_M_DE**, towards selecting the most appropriate network to attach.

Step 1: **FUNC_LEVEL_MOB_M_DE** gathers from its corresponding protocol level DEs (i.e. **PROTO_LEVEL_MIPv6_DE** and **PROTO_LEVEL_PMIP6_DE**) the information of the available access networks in the user's locality and from **NET_LEVEL_MOM_DE** the corresponding networks' policies and then, informs/triggers node's **FUNC_LEVEL_QoS_M_DE** in order to make the decision of the most profitable network to be attached to, in terms of user's service QoS prerequisites fulfilment.

2: **FUNC_LEVEL_QoS_M_DE** gathers a) information of node's/user's required service QoS prerequisites, b) information broadcasted by the cells in its locality (CDMA or WiFi) and c) the information provided by **FUNC_LEVEL_MOB_M_DE** and then, selects the most appropriate network the node should attach to (via setting and solving a corresponding optimization problem).

Via such a simple scenario the following will be demonstrated:

- The basic autonomic components of the proposed architecture, the design in line with GANA and the interactions and synchronization towards overall decision making, via the interfaces defined by GANA.
- The ability of the autonomic node to select the most appropriate network to attach to in contradiction to current architecture where such decisions are taken in a centralized way.
- The advantage of such a decision (even if it is taken in a decentralized way)

Scenario's 2nd Part: A roaming users – Mobility-triggered QoS-aware vertical handoff.

*Action: An autonomic mobile node (User 2) is roaming over a heterogeneous wireless environment and is currently attached to a CDMA cell (A), while an ongoing real-time service is provided to this user. A connectivity- triggered handoff is initiated. Thus, **FUNC_LEVEL_MOB_M_DE** indicates low signal strength and initiates handoff.*

Such a scenario aims at demonstrating some of the fundamental novelties and differences of the proposed autonomic architecture with respect to the currently existing ones (i.e. state-of-the-art approaches in the area of QoS-aware mobility managements).

Traditionally, following conventional approaches, since the mobile node is attached to a CDMA cell (A) its handoff to CDMA cell B would be performed. In this case such an option is not optimal since cell B is overloaded. However, in our case the following actions would take place instead:

Step 1: **FUNC_LEVEL_MOB_M_DE** peers with **FUNC_LEVEL_QoS_M_DE**.

Step 2: **FUNC_LEVEL_QoS_M_DE** gathers specific information constantly broadcasted by the base stations in users' locality via **PROTO_LEVEL_AP_R&Q_WLAN_DE** and **PROTO_LEVEL_BS_R&Q_CDMA_DEs**.

Traditionally, if a conventional myopic network selection strategy was applied (where a real-time/multimedia service would select a CDMA cell while a non-real-time data service would select WiFi service), then, the mobile node should be attached to CDMA cell B, which is not the optimal solution. However, in our case the following actions would take place instead:

Step 3: **FUNC_LEVEL_QoS_M_DE** sets and solves a multi-criteria optimization problem due to the information included in the normalized parameters (information) broadcasted by cells' base stations that reflect:

1. All cell's load and QoS-support conditions.
2. All cell's signal strength.
3. User's QoS-requirements fulfillment if attached to these cells

Step 4: **FUNC_LEVEL_QoS_M_DE** selects the WiFi and then peers with **FUNC_LEVEL_MOB_M_DE** to inform it for the target network that the upcoming handoff shall take place.

Step 5: **FUNC_LEVEL_QoS_M_DE** enables the autonomic radio resource management mechanism (ARRM) in the WiFi (i.e. **PROTO_LEVEL_NODE_R&Q_WLAN_DE**) and disables the corresponding one of the CDMA network (i.e. **PROTO_LEVEL_NODE_R&Q_CDMA_DE**).

Step 6: **FUNC_LEVEL_MOB_M_DE** orchestrates **PROTO_LEVEL_PMIP6_DE** and **PROTO_LEVEL_MIPv6_DE** towards performing the handoff.

Step 7: **PROTO_LEVEL_PMIP6_DE**, collocated in the base stations and access points, performs the fast handoff procedure between the involved CDMA base station and WiFi access point.

Scenario's 3rdPart: QoS-triggered vertical handoff.

Action: An autonomic static mobile node (User 3) currently attached to the WiFi access network, which becomes overloaded due to User's 2 attachment performs a vertical handoff to CDMA cell A, towards improving its service performance, since:

a) cell A is lightly loaded

b) the user interacts with his mobile terminal (node) and requests an enhanced real-time service with more demanding QoS criteria (than its current).

Via the proposed scenario we aim at:

- i) revealing autonomic mobile node's self-optimization functionality, enabled by the proposed architecture, that allows the user to initiate and perform a vertical handoff due to QoS-related reasons (even if the user is static).
- ii) demonstrating proposed autonomic architecture's enhanced flexibility towards supporting advanced autonomic functionalities, like QoE.
- iii) presenting integrated network's increasing performance benefits via load balancing.

Step 1: **FUNC_LEVEL_QoS_M_DE** gathers specific information constantly broadcasted by the base stations in user's locality via **PROTO_LEVEL_AP_R&Q_WLAN_DE** and **PROTO_LEVEL_BS_R&Q_CDMA_DEs**.

Step 2: **FUNC_LEVEL_QoS_M_DE** of the mobile node interacts with the user and interprets its desire for high definition video (currently requiring low quality video) to appropriate QoS metrics (i.e. identified the new required service class). Then, it sets and solves a multi-criteria optimization problem.

Step 3: **FUNC_LEVEL_QoS_M_DE** selects the CDMA cell and then peers with **FUNC_LEVEL_MOB_M_DE** to inform it to initiate a vertical handoff, as well as for the target network.

Step 4: **FUNC_LEVEL_QoS_M_DE** enables the autonomic radio resource management mechanism (ARRM) in the CDMA network (i.e. **PROTO_LEVEL_NODE_R&Q_CDMA_DE**) and disables the corresponding one for the WiFi (i.e. **PROTO_LEVEL_NODE_R&Q_WLAN_DE**).

Step 5: **FUNC_LEVEL_MOB_M_DE** orchestrates **PROTO_LEVEL_PMIP6_DE** and **PROTO_LEVEL_MIPv6_DE** towards performing the handoff.

Step 6: **PROTO_LEVEL_PMIP6_DE**, collocated in the base stations and access points, performs the fast handoff procedure between the involved CDMA base station and WiFi access point.

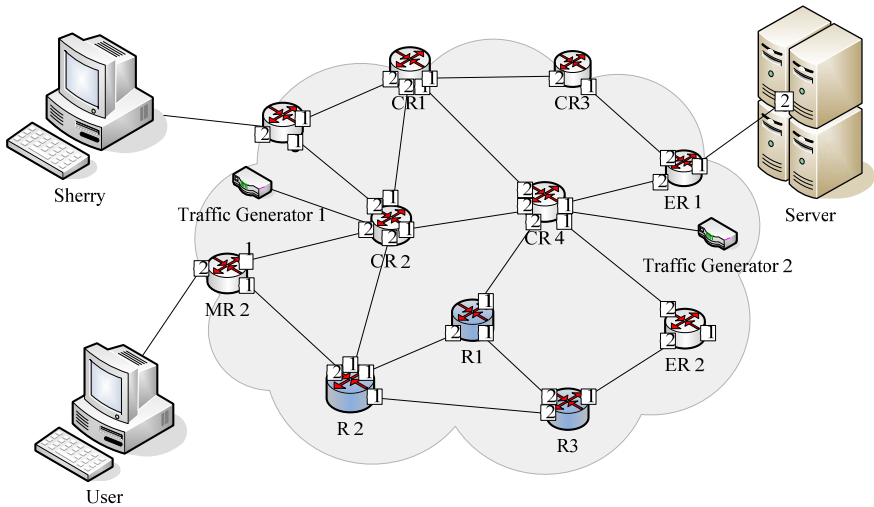
A-1.2.2 Autonomic QoS Management in Wired Network

The objective of this scenario is to highlight and demonstrate the functionalities of the autonomic QoS management over wired core environment as proposed and developed within the framework of project EFIPSANS (WP3). In this scenario the emphasis is placed in the demonstration of the self-adaptation and self-configuration functionalities of an autonomic node, which are enabled by the proposed architecture, and therefore reveal the potential emerging benefits from users' and networks' perspective.

High-Level Description of the Scenario

Autonomic QoS Management over Wired Core Environment	
<p>The Story-line</p>	<p>Today, Video On Demand (VoD) is one of the most popular entertainment. It requires a mass of resource. With the number of users grows up, the network can be congested. In our program, auto-management attributes is added in, and it provides every user who pays to get a higher quality service a USB key, which contains a unique and encrypted key, and the key contains the user's profile (i.e. user degree) with which the system can self-adeptly identify the service level and self-configure the network to provide the corresponding service according to the degree within the profile. The USB key can be attached to some portable device, e.g. Flash Disk, and at the time the device is plugged in, the system can identify the user profile, and then do a lot of things to self-configure the network, which can effectively improve the performance of it.</p> <p>Video on demand is one kind of the video service, meaning that playing the corresponding video program according to the needs of users, and the user can watch the direct-on-demand content without ads and not in a hurry for a program. It fundamentally changes the way that users watch television over the past passive deficiencies. And Sherry is the one.</p> <p>Sherry is a movie-liker, in deep love with VOD to see a movie. On working days, there are few people on the net, so the network is OK, and the common VOD service providing is good enough for users. When it is at weekend or at night, the peak of surfing, the network resource would be inadequate, and then the video will in the state of jammed, which is out of the tolerance of Sherry. So Sherry applies the VIP member, and receives the VIP USB key which contains Sherry's identity. After plugging-in the USB, the quality of the video increases obviously. Having higher quality of service, Sherry is very satisfied.</p> <p>For using the VIP service costs some money, Sherry do watching directly without USB key when the network is OK. If the state of the network becomes terrible, Sherry would plug-in the USB to continue, and at this time, the system would identify the USB key immediately and do some configuration, then provide Sherry her corresponding service.</p>

Short description of the scenario	This scenario illustrates the key functionalities of the autonomic QoS management architecture over wired environment. Our goal is to demonstrate some basic self-* functionalities of an autonomic node which are enabled by the proposed architecture, as well as to reveal the emerging benefits from both users' and networks' perspective.	
Scenario Scenes	1 st Part: A VIP member using a USB key is requesting a VOD service –Self-configuration of the network 2 nd Part: Plug-in the Flash Disk when watching the video –Self-adaptation to the change of the network	
Current problems/limitations with current practices and/or current technology		
Network Environment	Wired (core) Network	
Self* Functionalities introduced	Self-*Functionality	Problems/Limitations it addresses
	<i>Self-Adaptation</i>	<u>Current Practices:</u> Current practices concerning self-adaptation on wired (core) environments are limited and mainly based in conventional myopic criteria such as service differentiation.
	<i>Self-Configuration</i>	<u>Current Practices:</u> The lack of self-configure capabilities in current networking framework results in the node of the network cannot identify the different users and configure automatically the different service for different user.
Self-Adaptation – What it solves and the benefits	Self – adaptation of autonomic nodes when plugging in the Flash Disk to enjoy the VIP priority over a busy and varied network can be realized by a) constantly monitoring various parameters of the changing environment b) processing, combining and assessing such information, c) reacting on specific triggering events by choosing more suitable actual protocol to accomplish QoS guarantees, which should more fit the current tackled service flow.	

Self-Configuration – What it solves and the benefits	<p>Self-configuration function will</p> <p>a) be implemented in the advent of the network circumstances' changing.</p> <p>b) meet the current network circumstance, the protocol level DE would automatically configuration/ reconfiguration the parameters of the QoS mechanisms what they configured respectively,</p> <p>c) and as a result, the users would gain the optimum performances in the different application scenarios.</p>	
System(s) Involved	 <ol style="list-style-type: none"> 1. One Server which is used to response sherry's demand to send video to her. 2. Several routers to make up the wired network, while some of them connect Sherry's PC to server 3. Two PCs, one is Sherry's, which she uses to watch the video, and the other is used to act as other users who are in the same field to request the video at the same time too. 	
Key players that benefit	Actor/Player	Benefits (with Rationale)
	Operator	Revenue maximization and optimal utilization of the network resources under given QoS constraints in wired networks while achieving overall system's load balancing.
	Manufacturer	None
	End User	Achieves higher services' QoS experience in terms of quality and performance.

List of cooperative DEs and functionalities (Abbreviations)

During the evolution of the proposed scenario, the following DEs will communicate, interact and collaborate, which reside at the autonomic nodes:

No	DE's Acronym	DE's Name
1	NET_LEVEL_QoS_M_DE (Fixed Networks)	Network-Level Quality-Of-Service-Management-(Fixed-Network) Decision-Element
2	FUNC_LEVEL_QoS_M_DE (Fixed Networks)	Function-Level Quality-Of-Service Management (Fixed) Decision-Element
3	PROTO_LEVEL_SA_DE	Protocol-Level Service-Aware-In-An-Edge-Autonomic-Node-Of-The-Wired-Network Decision-Element
4	PROTO_LEVEL_PM_DE	Protocol-Level Packet-Marking Decision-Element
5	PROTO_LEVEL_QM_DE	Protocol-Level Queue-Management Decision-Element
6	PROTO_LEVEL_QS_DE	Protocol-Level Queue-Scheduler Decision-Element

GANA Ontology oriented Scenario Description

In the following a detailed description of the proposed scenario is provided. The scenario includes three different parts in order to better demonstrate specific features and autonomic functionalities. Moreover, for each part we outline the sequence of actions that will take place, as well as the involved corresponding DEs.

Scenario's 1stPart: A VIP member using a USB key is requesting a VOD service – the Self-configuration of the network

Action: After plugging-in Flash Disk which with USB key attached, Sherry, the VIP member is requesting a VOD service.

The purpose of this part is twofold. On one hand, it aims at illustrating the key components/DEs of the architecture for QoS management in wired core networks and the actions/decisions that the corresponding control loops are responsible for, while on the other hand, it will demonstrate the decision making operation of different kinds of autonomic nodes' FUNC_LEVEL_QoS_M_DE in the network and the self-configuration of protocol-level DEs, towards selecting the most appropriate protocols and parameters to get network resources best utilized.

Step 1: At the source end of the service flow, it pre-marks the service flow, and sends the pre-marked service flow along with the sensed user's profile to edge node.

Step2: At the edge node, the FUNC_LEVEL_QoS_M_DE gathers from its corresponding protocol level DEs (i.e. SA_DE) the information of service (i.e. service type) and the state of real-time network and the

node, and then, analyzes the information and determine whether to change the combination of protocols, which is to satisfy the functional level QoS requirements and is for the optimization of the network.

Step3: According to the decision of FUNC_LEVEL_QoS_M_DE, the protocol level DEs ((i.e. PROTO_LEVEL_SA_DE, PROTO_LEVEL_PM_DE, PROTO_LEVEL_QM_DE and PROTO_LEVEL_QS_DE)) chooses appropriate algorithms and parameters to service aware, packet marking, queue management and queue schedule by collecting its interested information. In which, PROTO_LEVEL_PM_DE should sufficiently consider the results of the pre-mark operation and the user's profile to do appropriate marking, as the base of guarantee of higher QoS providing.

Step4: When the service flow arrives at the core node, the FUNC_LEVEL_QoS_M_DE collects the context information and its corresponding protocol level DEs' information from lower level DEs, and makes the best decisions. It chooses appropriate protocols of queue management and queue schedule as well as their parameters. The PROTO_LEVEL_QM_DE and PROTO_LEVEL_QS_DE works automatically to make sure that the IP packets can be delivered quickly and with lower drop probability

Step5: DEs at all levels collaborate together to achieve self-*functionalities of the network, at the same time provide higher quality of service to users.

Scenario's 2ndPart: Plug-in the Flash Disk when watching the video –Self-adaptation to the change of the network.

Action: When watching the video online, it could encounter the situation of network burst, and then the VIP member could request the high quality service just by plugging in the Flash Disk without stopping their work on hand. Then the system will automatically identify the user's Profile, trigger a series of mechanism to provide high-quality services, and then the video experience will improve significantly. With all of these the system could achieve the self-adaptation ability, self-configure the attributes of the network and self-optimize the network and the node resources.

Such a scenario aims at demonstrating the better performance the system provides than that without QoS management. Traditionally, when the network is busy, it is hard to provide good service to users. However, in our case we give the video packets higher priority to be dropped, and when the network is busy, it tries to protect them from being dropped to assure the service. The following actions would give the details:

Step 1: The moment the user terminal gets the USB key, it informs the server. While the server senses the change of the service priority, it pre-marks the flow, and then sent the user profile to edge node.

Step 2: FUNC_LEVEL_QoS_M_DE gathers a) information of node's/user's required service QoS prerequisites, b) the context information and c) corresponding protocol level DEs' information from lower level DEs, analyzes the change, and then from the view of functional level to make decisions, choosing the best combination of protocols suit for the situation.

Step 3: The next steps are the same as step3 to step5 in the first part of the scenario. All of this is to provide corresponding level service for the user with USB key.

Via such a simple scenario the following will be demonstrated:

- The basic autonomic components of the proposed architecture, the design in line with GANA.
- The ability of the autonomic node to select the most appropriate protocol to adapt to the status of the network.
- The advantage of such a decision.

A-1.3 Autonomic Network Management Scenarios (WP4)

A-1.3.1 Autonomic Peer-To-Peer (p2p) Network Monitoring Scenario

Introduction

This scenario refers to autonomic network visualization, performance monitoring and resource management in fixed and wireless networks. In wireless environments, mobile ad-hoc networks (MANETs) are being created from neighboring nodes and communication is established through IPv6 connectivity in 802.11 networks or Bluetooth connectivity in case of connection of mobile devices with Bluetooth capabilities. In fixed environments different LANs may be connected or disconnected and the provided monitoring services are updated in an autonomic manner. In each case, the network administrator desires to have a continuous view of the network topology and also information about the current network status (established links, available paths, and performance metrics) and the available resources.

We try to exploit p2p networks characteristics in ad-hoc autonomic networks that are designed based on the GANA architecture. We present how peer-to-peer (p2p) techniques may improve performance monitoring and resource management in autonomic networks, especially in MANETs, since they both share the key characteristics of self-organization and decentralization and need to provide services in a decentralized, dynamic environment. P2p overlay networks are created on top of the existing networks for providing advances services and reducing network complexity.

High-Level Description of the Scenario

Provision of Autonomic Services in self-configurable environments	
The Story-line	<p>Autonomic nodes join an ad-hoc network and automatically recognize their neighbors. After establishing communication, they join the p2p overlay network and they are able to store information on it, through the use of Distributed Hash Tables (DHT). They can also query and retrieve specific information from the network in a decentralized manner.</p> <p>When an autonomic node wants to provide a service, it requests the data that is associated with this service in the network and starts providing it. Each service is implemented through the cooperation of a large number of autonomic entities that interact by storing and retrieving meaningful keys across the overlay ring topology. By this way, we can achieve autonomic functionalities that emerge in a decentralized manner without explicit action or control.</p> <p>A set of nodes that are selected automatically in the overlay network are responsible for storing the available topological information and providing the visualization service (upon request). These nodes collect data from all the network nodes, compose the received information and thus are aware of the current network topology and the network changes. Furthermore, they can provide the current view of the network topology to any requestor.</p> <p>A node is elected to provide monitoring in network level. Each node saves performance monitoring data in the network. The elected node is responsible to collect the monitoring data, upon specific requests from the other nodes, and provide the requested information or take decisions for more efficient monitoring.</p>

	<p>The following autonomic functionalities are provided:</p> <ul style="list-style-type: none"> • <i>Self-configuration</i> of the autonomic node. Each autonomic node recognizes its neighbours, joins the overlay network and is able to store and retrieve data autonomously. • <i>Self-organization</i> of the network – establishment of the network and the communication links. Join and leave node requests are handled autonomously. • <i>Self-optimisation</i> – data and services are distributed uniformly among the participating nodes. The established overlay topology is maintained and updated after changes in the network. <p>The scenario is implemented under different topologies for wireless and fixed networks and the challenges faced in each occasion are described in detail.</p>
Scenario Scenes	<p>Overlay Topology Stabilization: An overlay topology (ring) is formulated by all participant nodes in the physical topology. The creation and maintenance of the ring is an autonomic functionality since nodes join and leave the network and the ring is maintained or updated. This procedure is conducted as described in the ring maintenance protocol. The number of hops required for the ring to be stabilized or for re-stabilization in case of new node joins or leaves are very crucial autonomic performance metrics.</p> <p>Furthermore, the impact on stabilization for various sizes of the ad-hoc networks (average degree and number of nodes) or the probability of link failures (changes in the topology) will be examined.</p> <p>DHT Adaptation: After the stabilization of the ring, data can be stored in the network and autonomic services can be provided. Several p2p protocols can be used for this reason. In this scenario we have selected to implement all the functionalities according to Chord [13]. The metric that will be examined is the storage cost while the average degree and the number of nodes of the network will be changing.</p> <p>Visualization: A visualization service is provided in the established network. Data is exchanged in an autonomic manner among the participant nodes and each one of them can provide the service when requested. Visualization data is stored in the p2p network with a specific predefined key (e.g. using the DHT to the following text “visualization data”). Therefore, all network topology data is stored in one node in the overlay network. For reliability reasons, the Chord [13] protocol may store the same information to more than one node. A node in the overlay network may retrieve all data, analyze them and create the visualization service.</p> <p>Performance Monitoring: Collection, storage and sharing of monitoring data has to be done in a distributed manner for (i) improving scalability, (ii) increasing flexibility in ad-hoc deployments, and (iii) avoiding single points of failure. Each node assesses the performance monitoring data of the links with its neighbors. Typical metrics are one way delay (unidirectional), packet loss and utilization. Performance monitoring data about the link from the node A to node B is stored either in the node responsible for the key $H(IP_B)$ with tag IP_A or according to a hash function with the following parameters: $H(f(IP_A, IP_B, metric))$.</p>

	<p>Performance monitoring data is stored in the p2p network and each node is able to take decisions according to these data. A node from the p2p network is elected to analyze performance monitoring data for the whole network. This node either passes through the p2p network (e.g. go through the overlay ring) or retrieves data for specific links accessing relevant nodes. This node retrieves data from the “visualization service” related topology information.</p> <p>Monitoring in Fixed Networks: We consider a topology where different LANs are interconnected through (Linux) routers. When new LANs are attached (e.g. a new network interface of the router is active), they are automatically discovered and the overlay ring that takes in account all the connected LANs is created/updated. IPv6 characteristics and multicast messages are exploited.</p> <p>All the nodes of a LAN share the same group id, as the router multicast the group id to all the nodes in the LAN. The group id is given by the router using a simple mechanism such as extended Router Advertisements (RAs). The group id may derive from the router interface id (MAC), IPv6 address of the router or the network prefix. A node (e.g. with the smallest id) is selected from each LAN (group) to perform monitoring with similar nodes in different LANs. This node is called “representative group node”. Monitoring traffic is exchanged among “representative group nodes”.</p> <p>When two LANs are connected, their rings are merged. Then, all group ids are stored in a server responsible for a predefined key. Representative group nodes query the predefined key to learn other groups in order to establish monitoring.</p> <p>Admission Control: Traffic flows are injected in the ad-hoc network and admission control mechanisms are applied in the edge nodes. These nodes are able to take decisions for the admission of a new flow according to the information that receive from the available monitoring data in the ad-hoc network. They collect routing information for the path that the new flow has to follow, they check whether the new flow can be served and they take appropriate decisions.</p>
<p>Current problems/limitations with current practices and/or current technology</p>	<p>The challenge that we face is to design and implement autonomic functionalities in a decentralized manner in ad-hoc networks through the exploitation of p2p techniques. Resources have to be combined in a distributed manner and the following characteristics have to be supported: scalability, decentralization, high availability of the provided services, fault-tolerance. A logical overlay topology will be created that interconnects all the ad-hoc nodes and the designed functionalities will be based on lookup queries to this overlay network. This network will be created through the exchange of messages from each node with its neighboring nodes. An important characteristic of this approach is the capabilities for adaptation to the changes in the networking environment and the decrease in the administrator’s burden.</p>

Network Environment	Wireless ad-hoc networks, fixed networks	
Self* Functionalities introduced	Self- *Functionality	Problems/Limitations it addresses
	<i>Self-configuration</i>	<u>Current Practices:</u> A manually defined centralized server is needed in order to provide networking services (e.g. visualization). The participant nodes have to establish communication with the server and register for the services. In our approach, human intervention is minimized.
	<i>Self-organization</i>	<u>Current Practices:</u> Assumptions made by conventional DHT approaches (stable network, long lasting connections, stationary peers, relatively high bandwidth, hierarchical structure – efficient underlay routing, efficient connection establishment, dedicated routers), when applied to MANETs, require continuous monitoring and intervention from the network administrator. This is due to the existence of unreliable nodes and links, and the changing network characteristics.
	<i>Self-optimisation</i>	<u>Current Practices:</u> Services are provided in a centralized way and when crucial nodes are disconnected, the data and the services that they were providing, become unavailable.
Self-configuration – What it solves and the benefits	Each node can acquire an address, recognize a neighbour and participate to the overlay network autonomously. Then, the node knows where to store and query data and can provide services in collaboration with other nodes.	
Self-organization and Self-healing – What it solves and the benefits	The ring stabilization procedure prevents any inconsistencies at the provision of autonomic services. There is also robustness against failures and frequent network changes, leading to reduced complexity for the network administrator.	
Self-optimisation – What it solves and the benefits	The ad-hoc network is continuously adapted after node failures, arrivals or departures. In case of failure of nodes that provide specific services, reallocation of responsibilities is realized. The complexity for the basic operations is kept low: search complexity ($O(\log N)$), storage complexity ($O(\log N)$), maintenance complexity ($O(\log^2 N)$).	

System(s) Involved	mobile devices (mobile phones, laptops), network entities (routers, end hosts)	
Key players that benefit	Actor/Player	Benefits (with Rationale)
	Operator	Autonomic visualization and monitoring of the network. Each node can provide autonomic services. Load balancing in the network (storage and provided services). Autonomic configuration and creation of the overlay topology.
	Manufacturer	
	End User	Auto-configuration, no feedback needed for the creation of the overlay topology, access to all data and services.

Detailed Technical Description of the Scenario

Problem Statement: Research in the area of distributed systems on fixed networks indicated that the most robust way of creating reliable decentralized systems is to rely on a p2p protocol. Such protocols can be considered as reference implementations of a Distributed Hash Table (DHT). A DHT is a structure that is collaboratively built by all participating nodes of a p2p system and provides a lookup service for resources that are published by these nodes. Although many reference implementations exist the problem is that these protocols have been created with the assumption of the existence of a reliable fixed network. Additionally these protocols are capable of handling dynamic incoming and outgoing nodes and dynamic registration and deregistration of resources but still with the assumption of fixed topology and minimum uncertainty.

The challenge that we face is to design and implement autonomic decentralized functionalities in ad-hoc networks through the exploitation of p2p techniques. Resources have to be combined in a distributed manner and the following characteristics have to be supported: scalability, decentralization, high availability of the provided services, fault-tolerance. A logical overlay topology will be created that interconnects all the ad-hoc nodes and the designed functionalities will be based on lookup queries to this overlay network. This network will be created through the exchange of messages from each node with its neighbouring nodes.

Upon this ring, a p2p protocol (in our case Chord) could be put into normal operation without lengthy stabilization. Chord ensures that in a circular-ordered set of nodes, insertion (of a key value pair) and retrieval (of a value given a key) is achieved with logarithmic performance, as far as exchanged messages are concerned. Efficient resource storage and lookup is provided to an application level developer through a high level API which adheres to the following pattern: *insert(key_x, value_x)* and *getValueOf(key_x)*.

Selection of Chord: Chord is a simple but powerful protocol which solves the problem of efficient data location. It is an efficient distributed lookup system based on consistent hashing. Its only operation is to map a key to the responsible node. Each node maintains routing information about $O(\log N)$ other nodes, and lookups are feasible via $O(\log N)$ messages. Therefore, Chord scales well with number of nodes what makes it an interesting application for larger systems. Chord continues to function correctly even if the system undergoes major changes and if the routing information is only partially correct [1]. Chord does not implement services directly but rather provides a flexible, high-performance lookup

primitive upon which such functionality can be efficiently layered. Its design philosophy is to separate the lookup problem from additional functionality. By layering additional features on top of a core lookup service, overall systems will gain robustness and scalability.

The Chord protocol uses SHA-1 as consistent hash function to assign a m-bit identifier to each node and each key. Consistent hash functions are hash functions with some additional advantageous properties, i.e. they let nodes join and leave the system with minimal disruption. The m is an integer which should be chosen big enough to make the probability that two nodes or two keys receive the same identifier negligible. The hash function calculates the key identifier by hashing the key, and the node identifier by hashing the IP address of the node.

Routing is done with the **Dynamic Source Routing protocol (DSR)**. DSR is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows the network to be completely self-organizing and self-configuring, without the need for any existing network infrastructure or administration. The protocol is composed of the two mechanisms of Route Discovery and Route Maintenance, which work together to allow nodes to discover and maintain source routes to arbitrary destinations in the ad hoc network. The use of source routing allows packet routing to be trivially loop-free, avoids the need for up-to-date routing information in the intermediate nodes through which packets are forwarded, and allows nodes forwarding or overhearing packets to cache the routing information in them for their own future use. All aspects of the protocol operate entirely on-demand, allowing the routing packet overhead of DSR to scale automatically to only that needed to react to changes in the routes currently in use.

Description of the Scenario according to GANA principles

The main Decision Element that takes part in the control loops designed in this scenario is the Monitoring Decision Element (MON_DE). Each autonomic node's Mon_DE manages monitoring Managed Entities (MEs), takes decisions for the policies that have to be applied in the network and provides appropriate information to the rest autonomic elements (e.g. QoS Decision Element). Hierarchical control loops are designed in function and network level in order to describe the provided functionalities according to GANA principles.

The following Managed Entities are defined:

- P2P Managed Entity (P2P_ME): Implements p2p mechanisms according to a selected protocol and provides information to the Mon_DE. The P2P_ME is managed by the Mon_DE.
- Visualisation Managed Entity (Vis_ME): Collects topology information from all nodes and provides the visualisation service. The Vis_ME is managed by the Mon_DE.
- Traffic Monitoring Managed Entity (TM_ME): Implements monitoring mechanisms or tools and provide relevant information to the Mon_DE. The TM_ME is managed by the Mon_DE.
- Admission Control Managed Entity (AC_ME): Implements the admission control algorithm and ensures that traffic admitted into a node or network will not violate specified QoS performance guarantees. The AC_ME is managed by the QoS_DE.

Stabilisation: The P2P_ME of each node is responsible for implementing the procedure of joining the overlay network (recognize neighbours, successor and predecessor nodes) and maintaining the created ring after nodes joins/leaves. The ring formulation is implemented through a controlled flooding mechanism

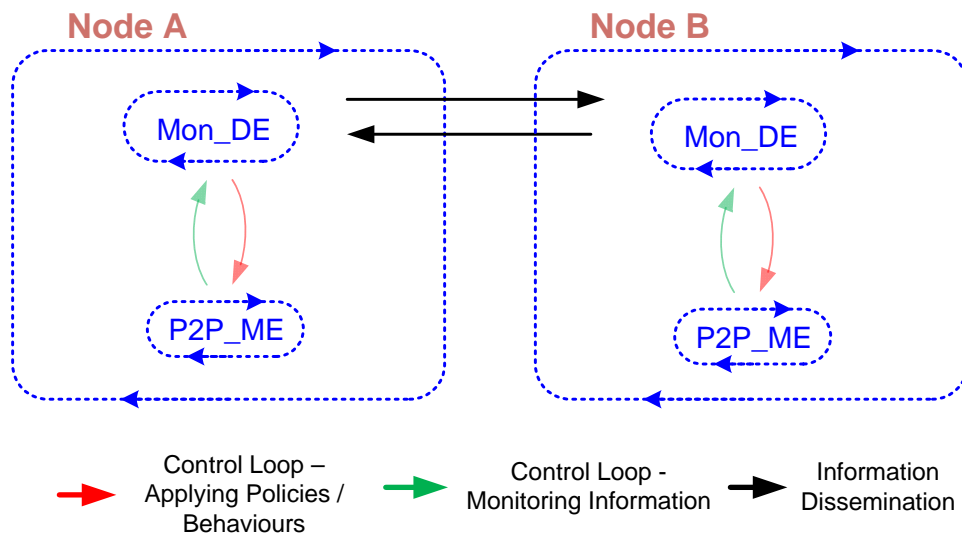


Figure 29: Setting up P2P overlay network though P2P_ME

Visualisation: The Vis_ME is responsible for neighbour discovery of each autonomic node and storage of the related information to the ad-hoc network, using a predefined key (e.g. “visualisation_key”). Control loops are designed for the provision of the visualisation service, as shown in the following Figure 30.

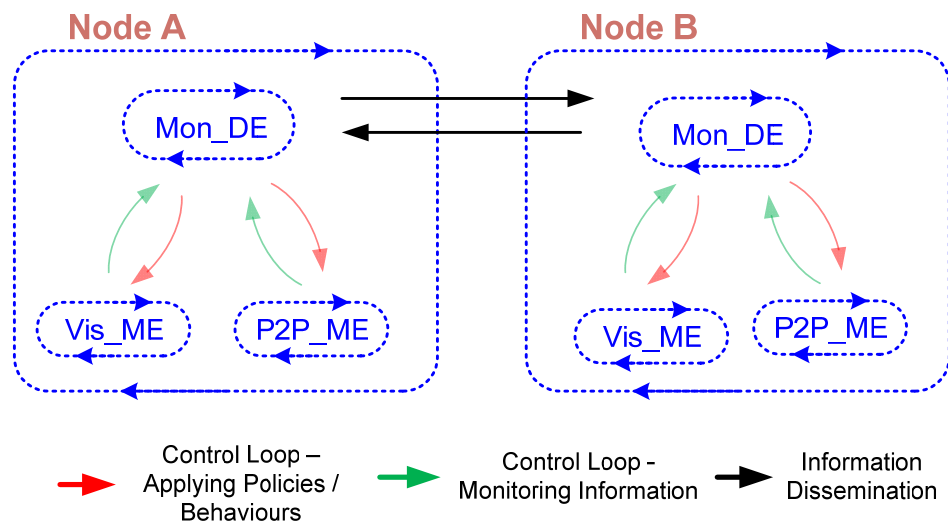


Figure 30: Neighbor Discovery through Vis_ME

Performance Monitoring: The TM_ME collects performance monitoring data for the links that are related with the autonomic node and the P2P_ME stores them into the overlay network. Upon request, the TM_ME can collect information for specific end-to-end links and facilitate the Mon_DE to take decisions (Figure 31)

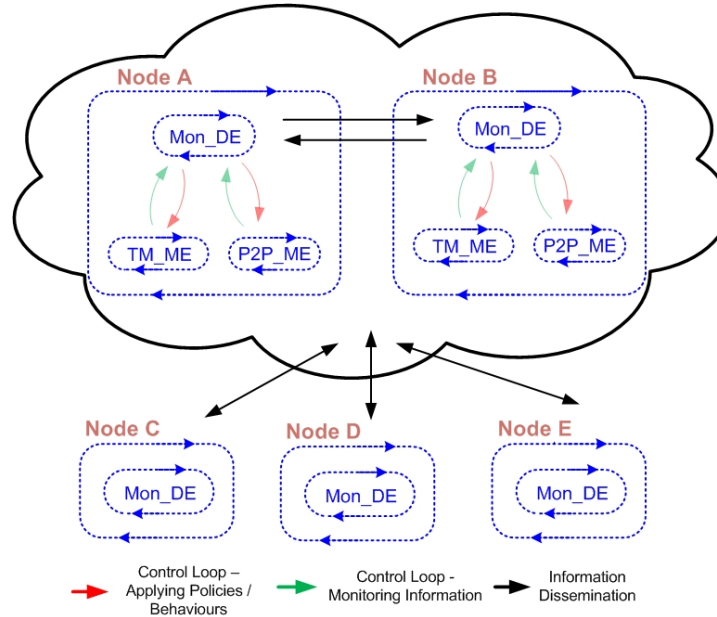


Figure 31: Performance Monitoring through TM_ME

Admission Control: The AC_ME accepts or rejects a request for a new flow among edge nodes in the ad-hoc network after receiving related information from the TM_ME. The autonomic behaviour is the ability to control the incoming traffic into a domain while maintaining a high degree of confidence in the admission decisions (Figure 32).

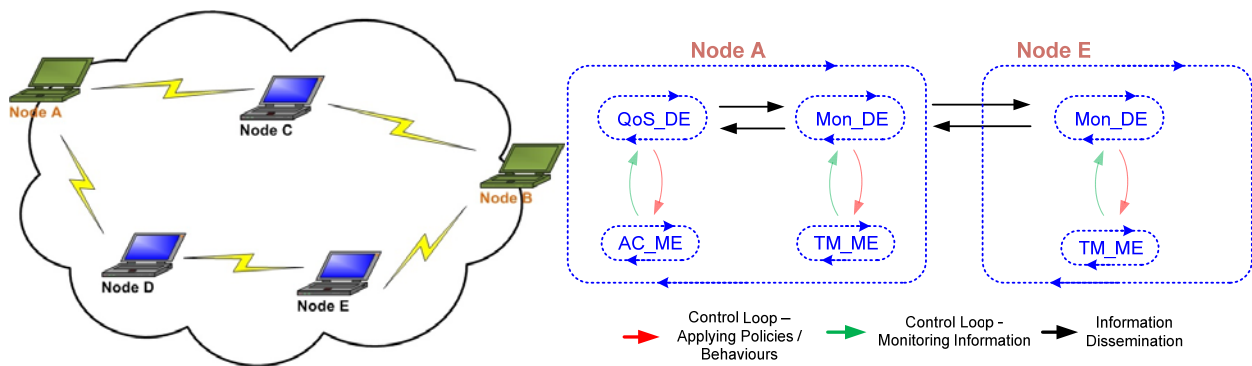


Figure 32: Admission control through AC_ME

A-1.3.2 Network Monitoring and QoS Management Scenario

In this scenario we discuss the interface requirements between specified GANA nodes and the network operator. The main scope of the following WP4 Wired Monitoring sub-scenarios is the clarification of FUNC_LEVEL_MON_DE operation, as an autonomic node's internal entity. We will try to approach its coordination role concerning its managed entities, its interaction with other DEs in the same node and finally the interaction with other autonomic nodes' DEs. Generally speaking WP4 Wired Monitoring Scenario focuses on autonomicity as a feature of traffic monitoring, coupled with Quality of Service (QoS) management functions, complying with GANA principles. As network and traffic conditions continuously change, monitoring protocols and mechanisms must be appropriately re-configured in order to facilitate the efficient QoS management of the autonomic network. In such complex environment we are examining our scenario from different perspectives with different functionalities.

In our first case, we presume that there is a node that handles network monitoring procedures. In this node network performance and services/application monitoring software should be installed and configured. Network performance and service/application monitoring software will constitute separate managed entities that will be orchestrated from the partially implemented FUNC_LEVEL_MON_DE. The sampling frequency and the type of the monitoring data that will be collected and locally stored are controlled from FUNC_LEVEL_MON_DE. The FUNC_LEVEL_MON_DE's decision should be based on the appropriate profile that the network manager will pass through the HCL architecture. It also includes interconnection between DEs which belong to the same autonomic node. From another more complicated perspective, we have added admission control which requires communication between DEs that belong to different autonomic nodes.

QoS Service Violation Case

In our first case Mon_DE flows perform several tests to different nodes and receive information about the status of each node. Monitoring tools will be installed in each node requesting for specific metrics and receiving an answer in a simple value format.

The Managed Entities that can be implemented are:

- Network Performance Monitoring: makes requests for specific metrics of the network. It sends a number of packets and retrieves a value for a specific test. In case of performance degradation appropriate alarms are being activated and log entries are being kept
- Services/Applications Monitoring: it monitors a set of services. According to different thresholds, severity and periodicity of network operation events that have been defined from network manager, the event handler of the FUNC_LEVEL_MON_DE decides what notifications should be generated. In case of recurrent events, or flapping metrics the event handler assumes responsibility to escalate logs and notifications messages generation.
- The event handler of the FUNC_LEVEL_MON_DE that will orchestrate the operation of the entire DE
- A local repository that will be used for monitoring data storage

Through effectors and sensory interfaces, we can load different policies in a policy registry in an Event Handler in the FUNC_LEVEL_MON_DE. For a number of applications/services there is a specific threshold where a notification must be sent in order for another entity or function to act.

The Decision Elements and Managed Entities that will be implemented are:

- Network Performance Monitoring: makes requests for specific metrics of the network. It sends a number of packets and retrieves a value for a specific test. In case of performance degradation appropriate alarms are being activated and log entries are being
- Services/Applications monitoring: it monitors a set of services. According to different thresholds, severity and periodicity of network operation events that have been defined from network manager, the event handler of the FUNC_LEVEL_MON_DE decides what notifications should be generated. In case of recurrent events, or flapping metrics the event handler assume responsibility to escalate logs and notifications messages generation.
- Event Handler: it monitors a policy registry, where services and applications are installed with specific requirements. It also, responds to a violation and informs through interfaces the according Decision Elements.
- Peer Interfaces: through these entities Monitoring DE can inform other Des, such as QoS_DE, in order to act in a service violation.

In Figure 33, we present a schema of the elements and entities that could fit in the above case. A schema that could translate all the above entities into a proposed architecture according to the GANA framework is shown below:

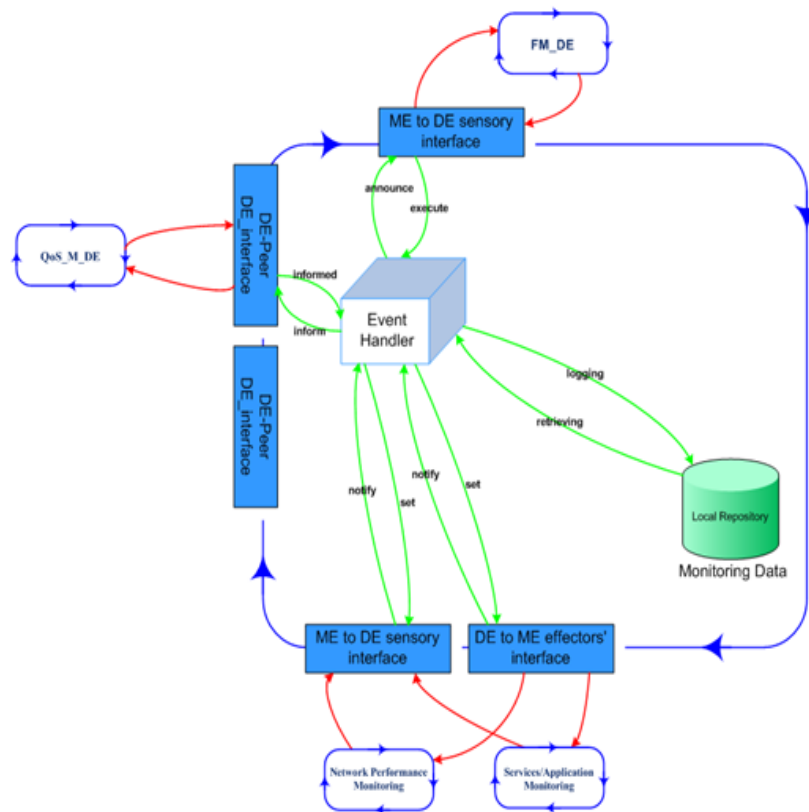


Figure 33: QoS Violation Scenario

Admission Control Case

Under our more complex case, one the following Decision Elements and Managed Entities will be implemented:

- **Effective Bandwidth ME (EB_ME):** implements an Effective Bandwidth estimation algorithm proposed in [14]. It collects a packet trace from the network, performs a number of processing activities on it and reports an estimation of effective bandwidth for a particular QoS target of packet delay.
- **Bandwidth Availability in Real-time ME (BART_ME):** This is an active probing process, used to generate and inject probe packets into the network towards a destination node. The BART_ME on the destination node estimates the amount of available bandwidth along the path between the two nodes.
- **Quality of Service DE (QoS_DE):** This is a function level Decision Element that participates to a node-local control loop. It aims to configure the mechanisms - such as queue management, queue scheduling, marking, policy, admission control, etc - to support service guarantees provided by the network.
- **Admission Control ME (AC_ME):** This Managed Entity ensures that traffic admitted into a node or network will not violate specified QoS performance guarantees. This process is based on the admission control algorithm presented in [15]. The AC_ME depends on measurements of effective bandwidth of the incoming traffic, realized by the EB_ME. The AC_ME has also an alternative configuration where high valued services may be given priority within the admission decision process, and that lower revenue services are throttled to maximise revenue for the service provider.
- **QoS Violation ME (QoS_V_ME):** It produces estimations of performance violations for traffic exiting the network through an egress router. The violations are estimated by analysing short packet traces and results are compared with particular QoS targets.

In Figure 34, we present the flow of information between Decision Elements and Managed Entities within the context of the GANA Monitoring Framework. The EB_ME supplies effective bandwidth information to the AC_ME, thus acting as an information supplier of that Managed Entity. If the QoS_DE detects that effective bandwidth estimation on admitted traffic needs to be updated quicker to compensate for the increased number of service request arrivals, it requests from the FUNC_LEVEL_MON_DE to re-configure the EB_ME accordingly.

The BART_ME supplies information regarding the bandwidth availability between the ingress node and the egress node to the QoS_DE on the ingress node. As there is best effort traffic traversing the network, congestion can occur within the core network. If congestion reaches a limit, this can affect the services that can be admitted while maintaining QoS targets of the admitted traffic flows. The QoS_DE therefore has a policy to reconfigure the AC_ME to prioritize higher revenue services during such times of congestion within the network.

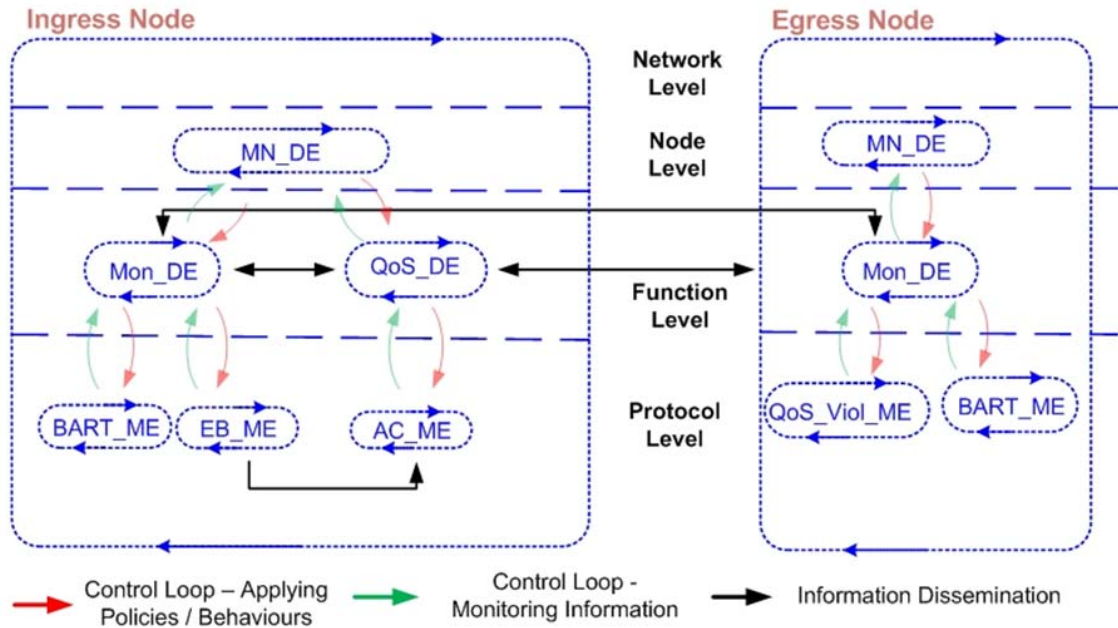


Figure 34: Admission Control Scenario

Autonomic Functionalities

Under this scenario, we have provide specific autonomic behaviors within the decision elements to ensure that, based on various changes in an autonomic network environment, such as traffic congestion, services violation, the DEs can configure the appropriate Managed Entities to operate more effectively under the new conditions.

The first case focuses on the requirements of the network operator to monitor these operations and the changing network conditions in the network. At each level of GANA, the network node can report information related to the hosted information suppliers, the managed entities and the decision elements

In our case with admission control, the autonomic behavior instilled within the ingress edge node is the ability to control the incoming traffic into a domain while maintaining a high degree of confidence in the admission decisions. This is provisioned by an interaction between the FUNC_LEVEL_MON_DE and the QoS_DE, where the traffic monitoring requirements of the AC_ME change and the associated EB_ME must be re-configured dynamically in order to conform to these changes. There is a dependency relationship between the lowest level MEs, i.e. EB_ME and AC_ME, of the ingress router, necessary for MEs to operate in an optimal manner. It is the responsibility of the interacting control loops, managed from the QoS_DE and the FUNC_LEVEL_MON_DE, to drive the re-configuration of these lowest level MEs.

Topology Information

This scenario will define a number of interacting Decision Elements (DE) across a network with the aim of managing QoS of admitted traffic, while adapting the behavior of various managed entities as conditions within the network change. This scenario is referring wired IP networks, where advanced monitoring information can facilitate the efficient provision of QoS. Network flows will be created between the nodes (IPv6 enabled) and will be monitored. The supported evaluation test bed topology is shown in Figure 35.

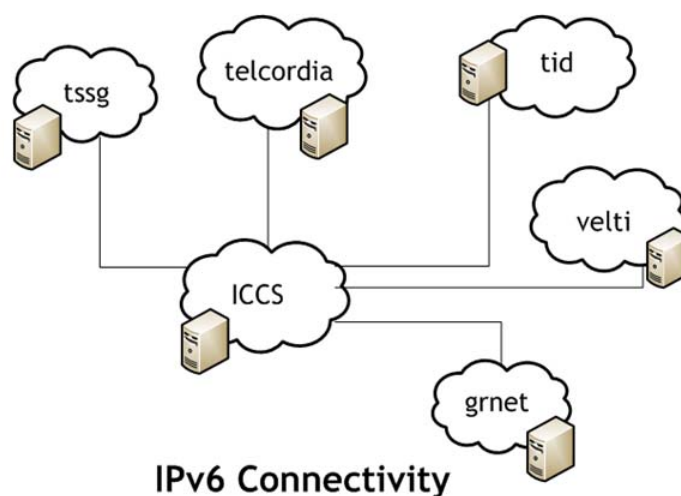


Figure 35: Testbed topology

Implementation Details and Specifications

Specific monitoring tools should be installed in each IPv6 node. Bandwidth Controller (bwctl) and One-Way Ping (owamp) run periodically in each node, collecting data and measurements for specific metrics. Bandwidth Controller [11] can measure maximum TCP bandwidth with various options available. One-Way Ping [12] is an application and policy daemon that is used to determine one-way latencies between hosts. Service monitoring can be succeeded combining NAGIOS [17] monitoring platform, as the basic element of the monitoring mechanism, and developing innovative monitoring plugins.

Bandwidth, delay, jitter and packet loss are the main metrics that should be measured and collected in a data repository. A framework that can analyze and show the metrics in a more visual way has been developed. That way the network operator can have a better view of any anomaly in the network.

The specified DEs and MEs that are referred in the general description of the scenario will be developed and will be applied to the edge nodes of the network.

A-1.3.3 Autonomic Fault-Management Scenario

Scenario Full Name: Autonomic Fault-Management and Reactive Resilience: “Autonomic Fault-Management for selected types of Black Holes in a fixed Network”

Since Reactive Resilience and Autonomic Fault-Management are converging concepts in an Autonomic Network, we intend to have a joint scenario demonstrating the benefits of realizing them in the context of GANA. This document starts with describing the general problem of a black hole that is obviously an issue for modern ISP networks and narrows down to a concrete case of black hole activation in a network, which is a potential candidate for the purpose of evaluating and demonstrating the Autonomic Fault-Management machinery implemented in the course of EFIPSANS.

High- Level Description of the Scenario

Autonomic Fault-Management for selected types of Black Holes in a Fixed/Wired Network	
Short description of the scenario	<p>This scenario aims at facilitating the GANA Autonomic Fault-Management components for the purpose of reducing the number of Black Holes in an operator's network. Black Holes are so called silent failures which are characterized by the fact that there is a physical and even logical connectivity between two (end-)systems, but data cannot be successfully forwarded between the two nodes (e.g. hosts).</p> <p>A concrete user oriented story line is described by the following steps:</p> <p>Joe is a passionate Internet user and often downloads big files via FTP or follows the world news over the video streaming services of the news agencies. However, it happens very often to him that FTP connections just get dropped, and that the video streaming just freezes, although in the same time he can browse, read emails, and is obviously connected to the Internet. The only thing that helps in such case is restarting the FTP download and/or the video streaming. Therefore, Joe complains often to his Internet Service Provider and is even thinking to withdraw from the contract and to use the services of another ISP.</p> <p>Bill is responsible for the network administration of the ISP's network, which provides Joe with Internet access. Bill receives many complains from customers and peer ISPs that traffic silently gets dropped inside his network and that many services provided to the end users are suffering because of that. Bill is concerned that he may not have the time and resources to investigate this problem in his network. Furthermore, the network operation personnel is forced to do many things in a rush, which results in bad configuration of equipment leading to unexplainable transient errors. As a result, the existence of Bill's company and a number of jobs is threatened.</p> <p>Bill creates a special task force inside his team that should try to find the causes for the packet delivery problems inside the network. After studying different research surveys and observing the events in the network, the task force concludes that the problems are very likely to be caused by different types of Black Holes due to misconfiguration of firewalls, bad routing protocol parameters leading to delayed convergences in routing, software bugs in the current stack implementations etc.</p> <p>Bill finds a vendor that provides equipment, which promises to cope with different</p>

	<p>types of problems related to IP black holes. The equipment is based on the GANA architectural model for Autonomic Network Engineering and is capable of resolving some types of Black Holes by means of Autonomic Fault-Management.</p> <p>The management of Bill's company decides that the last chance for them to survive is to purchase equipment from the aforementioned vendor and to install it in their network. After doing that the number of complains reduces significantly, and it turns out that indeed many misconfigurations , leading to black holes, inside the network are automatically removed by the GANA Decision Making Elements. Hence, Bill's ISP has been saved.</p> <p>In the mean time, Joe (the end user) has tried to change to another ISP, but the prices he has to pay there are not acceptable to him. Luckily, in the last weeks, after a number of complaints from his side, the connection breakdowns during download and video streaming have reduced. Hence, Joe is no longer thinking of shifting to another ISP.</p>	
Current problems with current practices and current technology	<p>Current devices do not intrinsically have any components that are able to exercise Autonomic Fault-Management and to resolve Black Hole problems. That is, whenever Black Holes occur it is very hard to detect the exact cause behind the faulty condition and to subsequently remove it. Black Holes are hard to detect since in many cases the connectivity is present and diagnostic tools like ping and traceroute fail to localize the problem, because the type of diagnostic packets they use may even reach their final destination without any problems.</p> <p>In general, Black Holes are likely to occur because of misconfigurations, software bugs, or as an emergent behavior resulting from delayed routing protocol convergence, etc.</p>	
Network Environment	Fixed/Wired	
Self* Functionalities introduced	Self-Healing/Repair	<p>Current Practices, Current Technology and their limitations:</p> <p><u>On Device's Level:</u></p> <p>As previously mentioned, the state of the current technology is that the devices are not equipped with components/facilities (e.g. GANA DEs) that can automatically solve the problem of Black Holes in an ISP's network.</p> <p><u>On Network Management Level:</u></p> <p>Given the transient nature of Black Holes, administrators are facing serious problems identifying and diagnosing the root cause for the erroneous state. Monitoring probes need to be instrumented in the network and different types of monitoring information</p>

		<p>must be correlated with the aid of the corresponding alarm correlation tools. Given that the root cause has been found, the removal of the Fault (root cause) is automatically conducted via the use of scripts.</p> <p><u>Limitations:</u></p> <ul style="list-style-type: none"> - The detection of the Black Hole is not a straight forward process because of the transient nature of Black Holes - Isolating the root cause and localizing the faulty device is time consuming and OPEX demanding task - The lack of components having the capabilities to resolve Black Holes, may lead to long lasting transient problems and breakdowns of connections in an ISP's network.
<p>Self-Healing/Repair – What it solves and the benefits</p>	<p><u>What it solves ?:</u></p> <p>The presented scenario for Self-Healing/Repair resolves some types of Black Holes in an operator's network. It relieves the human from conducting manually the processes of Incident-Detection, Fault-Isolation and eventually Fault-Removal with respect to Black Holes.</p> <p><u>Benefits:</u></p> <ol style="list-style-type: none"> 1) Black Holes will be removed during the operation of the network without causing any long-lasting problems – Robustness in the long-term operation. 2) OPEX reduction, since the overhead of humans detecting, diagnosing, and removing Black Holes is reduced 3) Increased availability of services to the end user, since some types of Black Holes will be removed within some reasonable time slices 	
<p>System(s) Involved</p>	<p>End-Systems, Routers - Edge Routers and Core Routers</p>	
<p>Key players that benefit</p>	<ol style="list-style-type: none"> 1. End-user 2. Operator 	

The detailed Technical Description of the INTERNET BLACK HOLES Scenario

The term *Black Hole* represents a family of packet delivery problems where the physical (and in some cases even logical) connectivity between two systems is present, but however the packets sent between the two nodes (e.g. hosts) don't reach their destination. The erroneous state results from the fact that the systems, even having the capabilities to react to the fault activation, do not get notified of packet delivery failures and cannot even localize the fault. That is, the sender may continue sending packets without detecting the packet loss problem and it cannot react. The forwarding nodes are also not aware of the problem and do not adapt their behavior correspondingly. The concept of black holes is described as follows:

"A number of complex failure modes exist that current systems fail to detect, and, therefore, are unable to recover from. In cases where the network is unable to automatically react to failures, manual intervention is required. Such action occurs on human time scales, however, and a potentially large number of customers may be offline for extended periods of time. Such failures are known as "silent failures" or "black holes" because existing networking protocols and devices do not alert on or automatically compensate for the failed component."

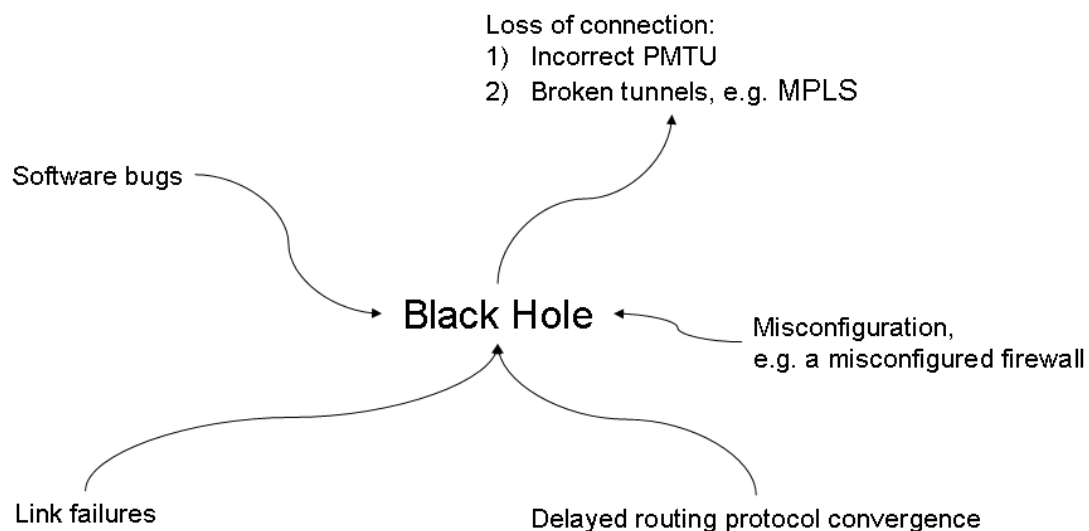


Figure 36: Possible causes for black holes

In other words, *when a packet enters a black hole it is lost, and there is no error reporting or alarm generated at the point of the black hole.*

The phenomenon of black holes has been extensively studied and its relevance for ISP operators is now well known. Some recent research work proposes advanced methods to detect black hole symptoms across a network and to diagnose a black hole activation. A discussion about handling black holes on the transport layer has been going on in the IETF, and has resulted in an RFC describing some aspects of the problem and proposing ways to handle them in the case of using TCP as the transport protocol. Furthermore, a real-time online service has been provided by the University of Washington that allows

monitoring black holes in the Internet and providing information about the AS they are originating from, their duration etc.

Figure 37 presents several possible causes for a black hole described in literature. Software bugs could cause the loss of packets on a router without an error message send back to the sending end system. The failure of a link and the delayed convergence of a routing protocol (e.g. OSPF) could lead to invalid routing paths in the network and to the loss of packets in the case of traffic flowing over the incorrect routes. Misconfigured firewalls, suppressing ICMP messages in an IPv6 network, could lead to systems using an incorrect PMTU after a link failure, since systems sending traffic would need to reinitiate the PMTU discovery process in order to get the correct one. Thus, packets are dropped on the router at which the PMTU has decreased because IPv6 does not allow fragmentation on routers for performance reasons. We elaborate on the last issue extensively in the next section. All factors, illustrated in Figure 36 can interplay in such a way that they lead to the “silent failures”/black holes, in which case the systems are not able to detect and localize the fault activation and respectively the protocols cannot adaptively react to it.

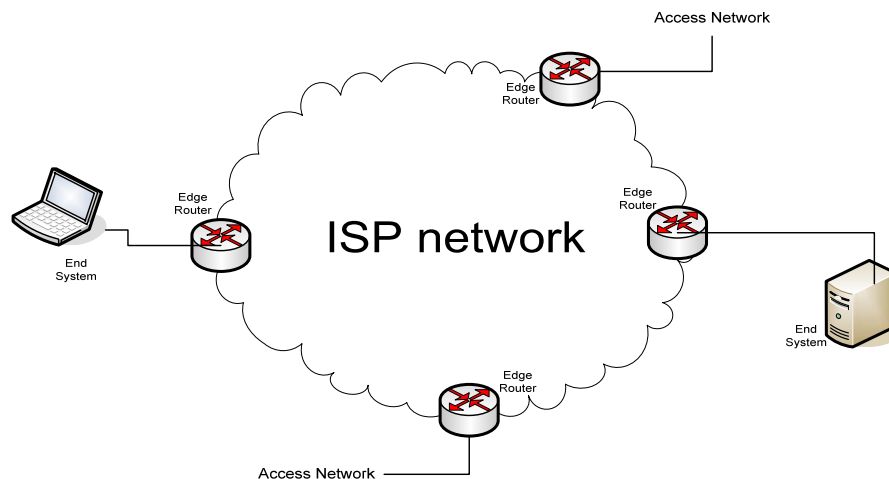


Figure 37: An Abstract Black Hole Scenario Setup

Figure 37 presents a high level view on the setup of the scenario for Autonomic Fault-Management and Reactive Resilience. Two end systems are attached to an ISP network and communicate over the routing paths established over the core and edge routers of the network. Given that a black hole gets activated inside the ISP network, either as an IP or MPLS black hole, the challenge is then to 1) *detect its symptoms*, 2) *perform Fault-Isolation and identify the root cause for the observed black hole symptoms*, and eventually 3) *reduce the impact of, or remove the root cause of the **black hole***. In parallel to the three steps, some *Fault-Masking actions* are required in order to preserve the already existing connections without disrupting the services offered to the end user. All these processes require to be mapped to the EFIPSANS architecture for Autonomic Fault-Management and Resilience, which is presented briefly in the Appendices.

The first step is the detection of a black hole. The relevant literature describes different active (probing) and passive approaches for the detection of black hole related symptoms. Several solutions propose to have specially instrumented probing components on selected nodes (e.g. edge routers) that periodically check different paths in the ISP's network for black hole routers. The information about the paths that are deemed to experience challenges is consequently correlated (e.g. by using bipartite graphs inspired approaches) in order to localize the faulty link and/or router. Some other papers describes the black hole related patterns that can be passively observed in TCP traffic flowing over a router, which is situated

between the sender end system and the router losing packets in a path. However, the related research does not propose methods to realize a chain of actions as the one in the following Figure 38. In other words there are no solutions to automatically minimizing the effect of black holes or removing them in an autonomic fashion. Our research, in the course of EFIPSANS, targets the realization of the overall set of processes in with respect to black hole activations. The *Fault-Removal* and *Fault-Masking* parts are the ones that have not been extensively examined in research up to now. Both of them could be either realized in the end-system (e.g. correcting the PMTU), on the *black hole* router (e.g. reconfiguration of a firewall suppressing ICMPv6 “*Destination Unreachable -Packet Too Big*” messages), or on a router inside an affected path by rerouting the packets or re-establishing a broken tunnel, e.g. MPLS tunnel. We consider that Fault-Removal is realized by the corresponding FM_DE (Fault-Management Decision Element) on the appropriate node/device and that Fault-Masking falls into the responsibility of the R&S_DE (Resilience & Survivability Decision Element) on one of the nodes being part of the affected path – either end systems or intermediate forwarding nodes. The detection of the black hole related symptoms is realized by specially instrumented monitoring component that after detection passes the incident description to the FDE, part of the GANA Dissemination Plane, which subsequently disseminates the information across the network and performs Fault-Isolation that is orchestrated by the FM_DE. The Network_Level_FM_DE needs also to be aware of the observed symptoms and to escalate and report the problem to the network administrator in the case that it appears not to be solvable by means of Autonomic Fault-Management.

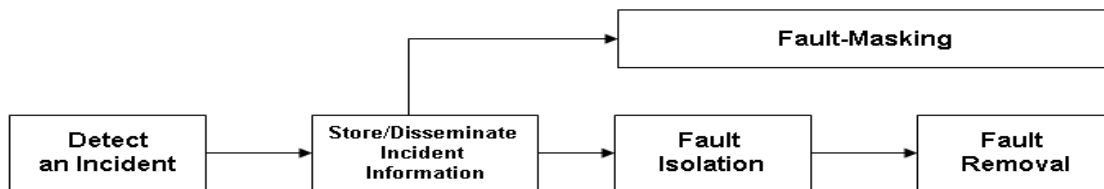


Figure 38: The Autonomic Fault-Management and Reactive Resilience in EFIPSANS.

Concrete Scenario Instance for Demonstration

Our ideas of a concrete scenario for demonstration are mainly based on the problem description. Besides the components for Reactive Resilience and Autonomic Fault-Management entities for monitoring and a Network Monitoring System, e.g. NAGIOS [17], are required to support the implementation of the scenario.

An IP black hole is a specific type of black hole that is caused by the loss of packets at a router, called a black hole router, without any (ICMPv6) error message being reported back to the end system sending the data. In the normal case, if a router drops a packet, it should generate an ICMP message informing the end system about the underlying problem. Consequently, the end system should adjust its corresponding PMTU in order to continue providing its services to the user. Since administrators often suppress some ICMP messages in their firewalls (security considerations), this problem can potentially occur whenever using old firewall configurations or the security concerns prevail inside an operator’s network.

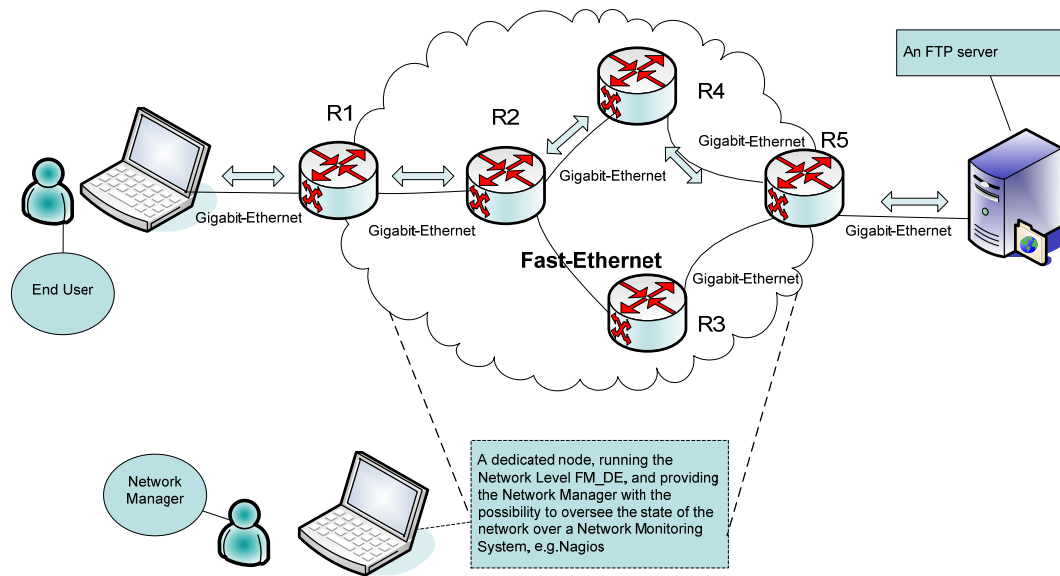


Figure 39: A testbed realizing the IP black hole Scenario

Given a network as the one in Figure 39 and assuming that R2 is such a misconfigured black hole router, one can think of a traffic moving between the FTP server and the end user system over a the path: **end system ↔ R1 ↔ R2 ↔ R4 ↔ R5 ↔ FTP server**. This path would also have an established Path MTU of the size allowed by Gigabit Ethernet jumbo frames (9000 bytes). Assuming that at a particular point in time, the link **R2 ↔ R4** fails (see Figure 40 below), then the traffic would get rerouted over the link **R2 ↔ R3**, which has the Fast-Ethernet MTU of 1500 bytes. Hence, the PMTU will decrease from 9000 bytes to 1500 bytes. Thus, R2 will start dropping packets because fragmentation is not allowed in intermediate routers in IPv6 (performance issues), and since R2 is configured to suppress ICMP messages, the corresponding end system won't get notified that it must readjust the size of the packets it sends out. This would normally lead to a time out of the connection on the upper connection-oriented protocols. In the case of TCP, the connection could survive given that TCP PMTU discovery is implemented and switched on, which is often not the case for the standard configuration of many operating systems. With Autonomic Fault-Management our goal is to have systems selectively employing different probing techniques to determine the correct PMTU at IP level, such that the system employs other means to discover the PMTU without relying on the native IPv6 mechanisms. Therefore, the actions Autonomic Fault-Management undertakes are of benefit for all transport layer communications including connectionless UDP. In the particular case, our goal is to preserve the connection between the end system and the FTP server.

Our work provides packet flow descriptions that illustrate a black hole as observed on an intermediate router, i.e. a router between one of the end systems and the black hole router. Such a router would be R1 in our case, because R2 is considered as the black hole router. Hence, a monitoring component can be implemented that is managed by the monitoring DE (MON_DE) and reports automatically the detection of a black hole to the FDE on R2. The IDE, part of the FDE on R2, would then disseminate the incident description to the end system and to the black hole router. The end system needs consequently to readjust the PMTU for the connection in question. This resilient mechanism is realized by the R&S_DE on the corresponding end system. The FM_DE on the black hole router R2 would need to reconfigure the local firewall in order to allow the appropriate type of ICMP messages and remove the root cause for the problem. Of course, the root cause would need to be identified first by the Fault-Isolation mechanisms in place, which are the Fault-Diagnosis/Localization/Isolation functions of the FDE instance on the black

hole router. Hence, the Fault-Isolation techniques require knowledge (e.g. in the form of fault/error/failure descriptions embedded in a Markov Model) that captures aspects from the configuration of the routers across the network, e.g. firewalls.

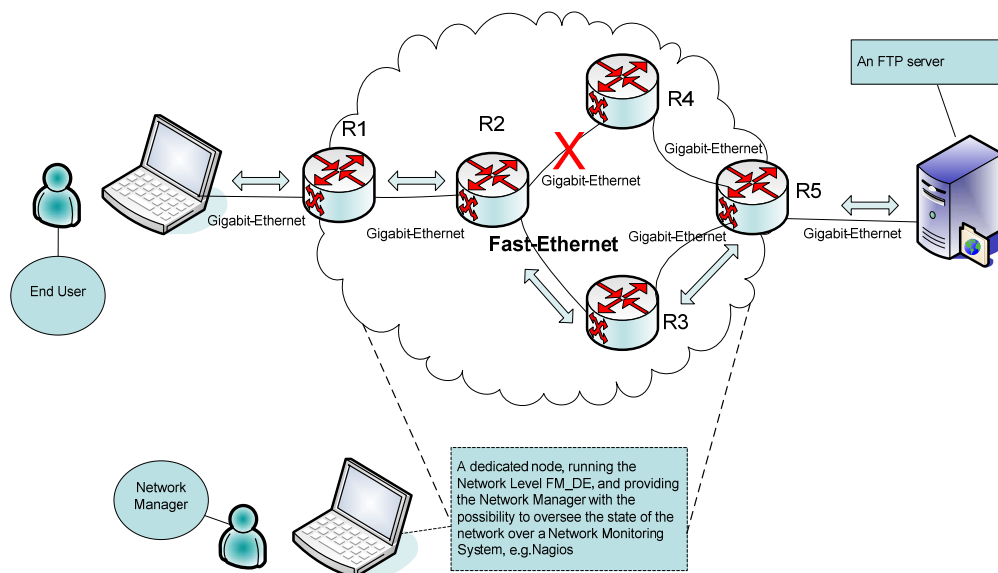


Figure 40: R2-R4 link is Down now the traffic is rerouted over a path with a smaller PMTU.

Other possible Scenarios

Apart from the scenario presented in the last section, other possible black hole problems could be considered for the following two cases:

1. Misconfiguration leading to a black hole that causes an MPLS tunnel to be broken
2. Delayed routing protocol convergence leading to a black hole