low Energy COnsumption NETworks

DELIVERABLE D6.4

LARGE-SCALE VALIDATION AND BENCHMARKING OF A NETWORK OF POWER-CONSERVATIVE SYSTEMS

Grant Agreement Number:	**258454**
Project Acronym:	**ECONET**
Project Title:	**low Energy COnsumption NETworks**
Funding Scheme:	Collaborative Project
Starting Date of the Project:	01/10/2010 *dd/mm/yyyy*
Duration:	39 months
Project Coordinator:	Name: Raffaele Bolla Phone: +39 010 353 2075 Fax: +39 010 353 2154 e-mail: raffaele.bolla@unige.it

Due Date of Delivery:	M39 *Mx* (31/12/2013 *dd/mm/yyyy*)
Actual Date of Delivery:	28/01/2014 *dd/mm/yyyy*
Workpackage:	**WP6 – *Integration, experiments and performance evaluation***
Nature of the Deliverable:	D
Dissemination level:	PU
Editors:	TELIT, CNIT, ALU, TEI, MLX, ETY, LQDE, INFO, NVR, NASK, WUT, GRNET
Version:	1.1

List of the Authors

TELIT	TELECOM ITALIA S.P.A
Diego Suino	
CNIT	CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI
Roberto Bruschi, Franco Davoli, Alessandro Carrega, Paolo Lago, Chiara Lombardo, Sergio Mangialardi, Fabio Podda, Mirko Rubaldo, Matteo Repetto, Antonio Cianfrani, Marco Polverini, Luca Chiaraviglio, Nanfang Li	
TEI	ERICSSON TELECOMUNICAZIONI
Renato Grosso	
ALU	ALCATEL-LUCENT ITALIA S.P.A.
Giorgio Parladori	
MLX	MELLANOX TECHNOLOGIES LTD - MLNX
Lavi Koch	
GRNET	GREEK RESEARCH AND TECHNOLOGY NETWORK S.A.
Anastasios Zafeiropoulos, Constantinos Vassilakis	
NVR	NETVISOR INFORMATIKAI ES KOMMUNIKACIOS ZARTKORUEN MUKODO RESZVENYTARSASAG
Árpád Bakay	
ETY	ETHERNITY NETWORKS LTD
David Levi	
INFO	INFOCOM S.R.L.
Maurizio Giribaldi	

Disclaimer

Copyright

Table of Contents

1 Executive Summary

The main target of the ECONET project consists of designing and developing a set of enabling technologies to reduce the energy requirements of wired network equipment, usually applied in Telecom and ISP infrastructures and at customer premises. This scope is obtained by working on dynamic power scaling, as a function of the real performance and service required to the device, and on smart standby, able to switch off the apparatus when no services are working, but also able to resume very quickly the functionalities when required.

The introduction of these energy consumption properties and functionalities into the control plane of networking equipment and services is enabled and eased by a uniform control interface like the Green Abstraction Layer (GAL), introduced in all devices developed in the project.

The realization of the final demonstrator represents the synthesis of the results obtained in the project. This demonstration reproduces a complete wire-line network chain from the core devices to the subscriber terminals, integrating quite a few of the green mechanisms proposed in Work Packages 3 and 5. The design of this network has been a challenging task, owing to the differences in terms of interfaces, protocols and dimensions of the various prototypes developed, and the different technologies involved.

The objective of this deliverable report is to describe the details of the demonstrator setup, trying to underline and to motivate the most significant design aspects, integration efforts, and testing environment. It is worth noting that the experimental results obtained with the demonstrator are not part of this document, but they are reported in the Deliverable 6.5.

The demonstration network has been designed in order to represent the most significant parts of a usual deployment of a medium scale Telecom Operator with both residential and enterprise network accesses, and including datacenter. In particular, the setup of the demonstrator includes 15 physical prototypes of network devices, emulated networks, and a number of hosts and traffic generators. This network can be seen as composed of six different network segments: home, access, metro, datacenter, core, and transport. All these parts are coordinated and managed by another essential element of the demonstrator: the green monitoring and management framework.

The scope of the demonstrator is not only to represent the prototypes and solutions developed in the project, but also to verify the quality of the achieved results with real tests and measurements.

Therefore, particular attention was also paid on the testing and simulation aspects and several profiles of traffic were used, in order to reproduce real working conditions on the devices in the network in the different times of the day and of the week. The traffic used during the demonstration has been fundamental to verify the power consumption performance of the network and of the devices, under appropriate Quality of Service (QoS) constraints, because this functionality is activated by the flow characteristics.

The results obtained by the measurements performed on the realised test bed prove the validity of the strategy of the project in order to reduce the energy consumption and of the technical solutions developed.

2 Introduction

The main goal of the ECONET project was to develop and realise prototypes and strategy solutions to obtain a significant reduction in energy consumption, without compromising QoS.

In addition to the standalone performance evaluation of each single prototype [1], it is fundamental to demonstrate that all the developed parts can be interfaced to build a complete green-enabled network chain, where heterogeneous green-enabled data-plane mechanisms, control-plane algorithms and protocols, as well as central monitoring and management framework can effectively interoperate and achieve significant reductions of energy consumption. A further objective of the demonstration is to obtain realistic experimental results capable of assessing not only that the integration of the several prototypes and technical solutions is working in an effective way, but that the goals of the project in terms of energy saving have been attained.

The aim of the present deliverable is to describe the demonstrator, its structure and topology, the green mechanisms integrated, the test strategy and the single segments that compose the test bed.

In order to disseminate this information as much as possible, the description of the demonstrator is supported with a number of live videos collected during the final demonstration test sessions. These live videos have been made publicly available in a new section of the ECONET website, which can be found at the following URL: https://www.econet-project.eu/portal/.

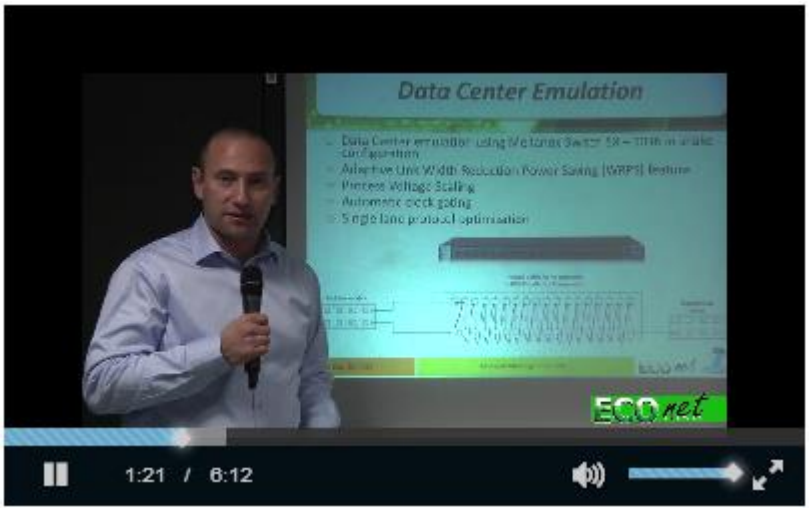
In more detail, the following selection of live videos is relevant to the present document, since it includes the introduction to the green technologies integrated in each ECONET demonstrator segment. (Note that each specific video can be accessed by means of the hyperlink onto each video screenshot):

- Optical Transport Cloud (ALCATEL-LUCENT):

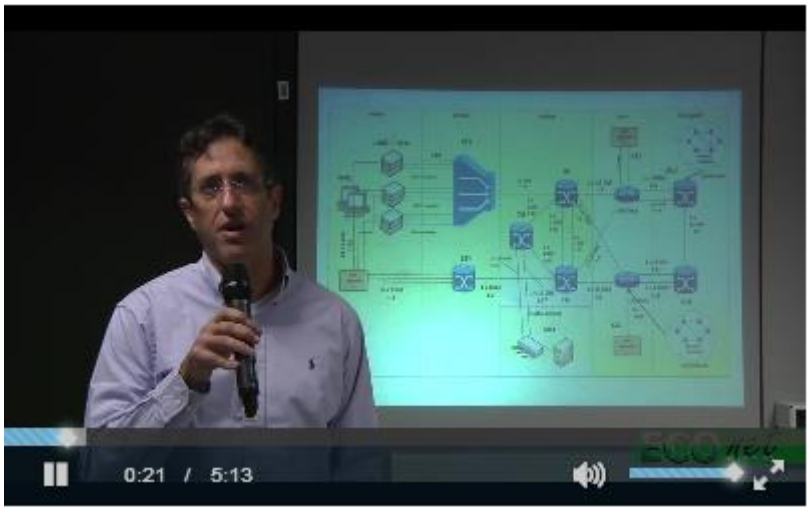
- Metro Transport Cloud (ERICSON):

- Core Cloud (CNIT GE-TO-RM):

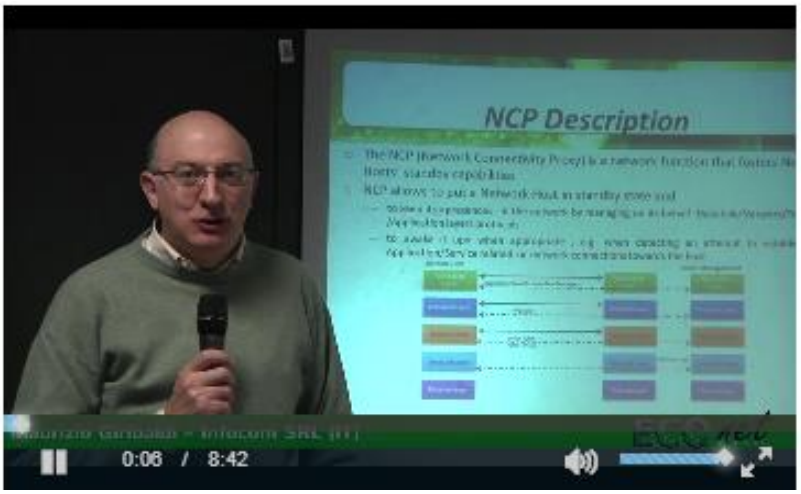
- Data Center Cloud (MELLANOX):



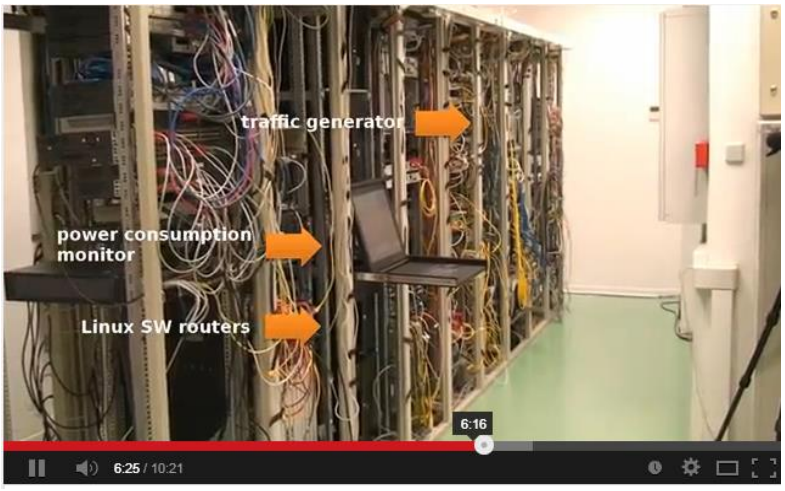
- Access Cloud (ETHERNITY):



- Home Network Cloud (LANTIQE and INFOCOM):



In addition, an appendix (Appendix B) reports the activities and the results obtained in a further small-scale demonstrator setup in the NASK facilities for validating local and network-wide policies. In this case too, a live video has been captured and published on the ECONET portal.



2.1 Structure of the Document

In addition to the Executive Summary and the Introduction (Section 2) the deliverable includes the following 6 sections:

Section 3, Topology of the test bed: this section introduces the architecture of the demonstrator and provides the general interconnection among the several parts composing the network and the role of each segment in the topology. A description of the parts of the emulated (and not physically reproduced) networks is also present.

Section 3, Traffic generation and profile: this section describes the strategy adopted to verify the performance of the network and of the devices in terms of energy efficiency. A detailed description of the different traffic flows used and their profiles as function of the hour of the day is given. Schemes and tables describe the flow parameters and the paths of the traffic flows in the physical demonstration network.

Section 4, Functional Architecture of the Demonstrator: this section describes the technique used for network control and management. In particular, The Green OAM framework and the general functionality of the Green Abstraction Layer (GAL) are presented, along with the new protocol developed in the project for remote power management, named GAL REpresentational State Transfer (REST) protocol.

Section 5, Parameters collected and Measurements performed: this section describes the parameters collected during the tests and the metrics used in the data analysis to evaluate the energy capability of the network and of the individual devices.

Section 6, Demonstrator layout: this section describes the single parts of the demonstrator and provides a detailed description of each sector of the network. In particular, the section is divided into 6 further subsections describing the details of the different sectors:

- Optical Transport.
- Core Cloud.
- Metro transport.
- Data centre.
- Access.
- Home network.

Section 8, Conclusions: this section concludes the deliverable with a summary of the organization and features of the demonstrator.

3 Topology of the test bed

The demonstration network has been designed in order to represent the usual deployment of a medium scale Telecom Operator with both residential and enterprise network accesses, including datacenter.

The design and the dimensioning processes of such demonstration network have been quite complex tasks, since the ECONET Consortium had to consider and to combine a number of "low-level" technological requirements and hardware availability issues.

Among the most critical points, it is worth mentioning that the demonstration network was built by heterogeneous prototypes coming from various ECONET partners, and integrating quite a few of the green mechanisms proposed in Work Packages 3 and 5.

In addition to the much different maturity level intrinsic to the prototypes (which directly affects their stability and the possibility of using them in some specific configurations), many of them have different types of interfaces (in terms of line protocols or protocol versions), and, in certain cases, some incompatibilities have been encountered. To this purpose, in many cases, prototypes have been completed with "legacy" hardware needed to provide interconnectivity or to realize topologies suitable to the usage of the ECONET control-plane algorithms and protocols.

Moreover, also the dimensioning of the available link bandwidth and of the number of interconnection links has been carefully decided. In fact, the result of the dimensioning process had to provide enough bandwidth to carry the traffic volume ranges suitable to trigger the various green mechanisms available in the clouds.

As shown in Figure 1, the demonstration network includes 15 physical prototypes of network devices, emulated networks, and a number of hosts and traffic generators. For the sake of simplicity, we divided the demonstration network into 6 "clouds," which represent different network segments, namely: home, access, metro, data-centre, core, and transport.

At a glance, the 6 demonstration clouds can be summarized as follows (further details can be found in section 7):

- *Home Network Cloud*: it represents the customer's residential network or small business network terminations. Each home network is composed by a VDSL HG and optionally some network hosts. Five home networks have been deployed in order to make evident the impact of the different HG power states, and to perform experiments on the Network Connectivity Proxy.
- *Access Network Cloud*: it includes two different devices, namely a VDSL DSLAM for residential or small business network terminations, and a Gigabit Ethernet switch for enterprise network access or back hauling of wireless terminations.
- *Metro Network Cloud*: it is composed by a typical ring topology of three metropolitan carrier Ethernet switches.
- *Datacenter Cloud*: this cloud contains the most representative building blocks of a data-center, namely, a server equipped with a high-speed network adapter, and a high-speed interconnection switch. Both the server interface and the switch work with the 40 Gigabit Ethernet standard.
- *Core Network Cloud*: this cloud contains two instances of the DROP router, an open-source prototype based on the Network Function Virtualization (NFV) scheme, envisaged by ETSI as the next-generation approach to router architecture. Two emulation networks, including a

number of virtual routers implementing green routing strategies and protocol extensions, have been interfaced with the two DROP instances.

- *Transport Network Cloud*: it is composed of two physical nodes and of an emulation network aimed at representing the optical backbone transport network. The green mechanisms available in this cloud have been integrated with the ones available in the DROP routers in the previous cloud.

Moreover, as will be discussed further on in this document, a seventh cloud (the "Monitoring and OAM cloud") is also physically present in the demonstrator.

From a general point of view, and as still shown in Figure 1, all the physical interconnection links have been realized with various versions of the Ethernet protocol (at 1, 10, and 40 Gigabit speeds) and with the VDSLv2 protocol for the DSLAM – Home Gateway interconnectivity.

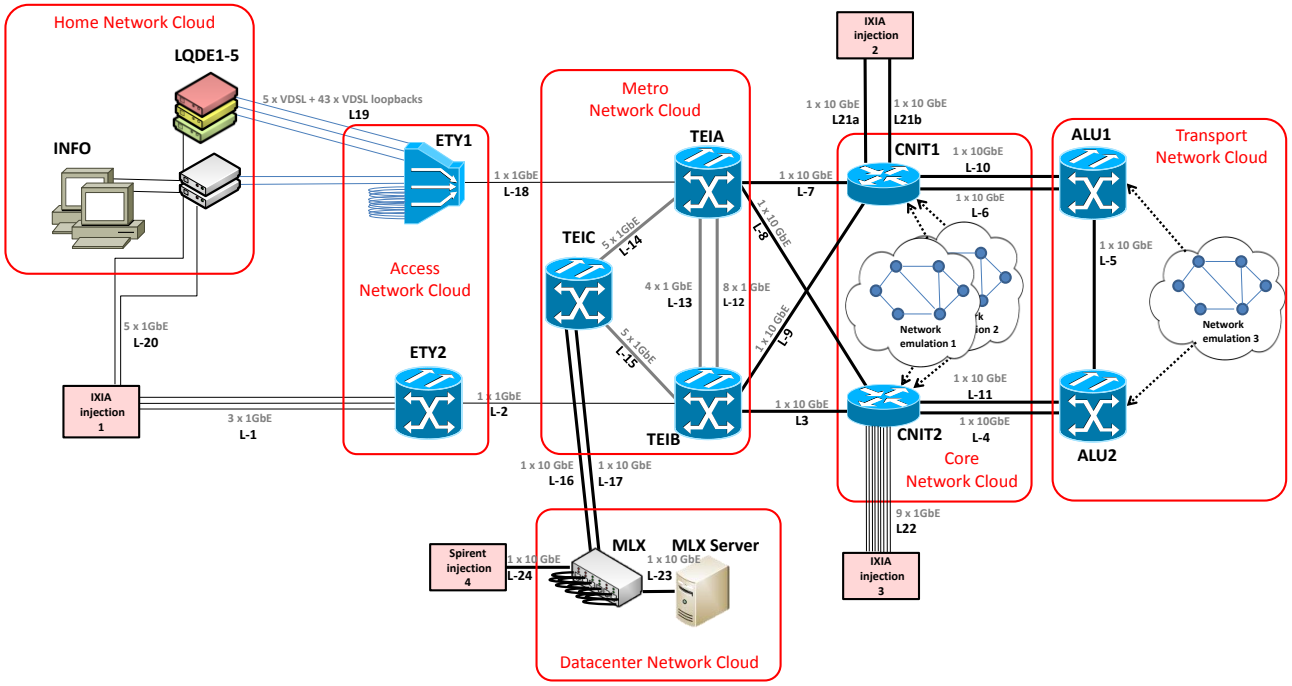


Figure 1: The ECONET demonstration network topology.

In certain cases, some specific solutions have been carried out to emulate the presence of additional devices, and to feed the existing prototypes with reasonable levels of traffic. For instance, we can mention that:

- To feed the DSLAM with suitable traffic levels, 43 VDSL lines have been put into the loopback mode;
- To feed the data-center switch with suitable traffic levels, a "snake" interconnection has been carried out (see Figure 2).

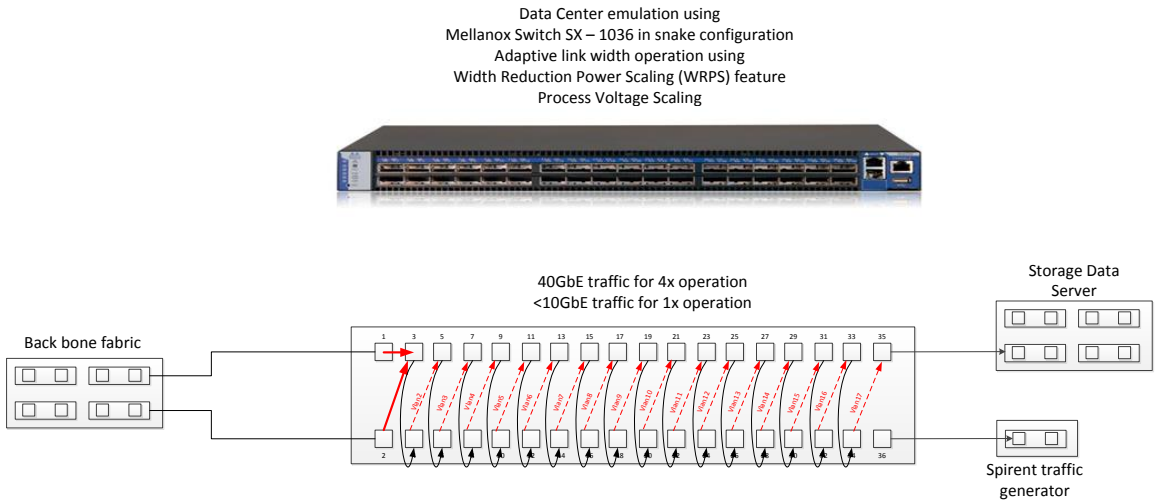


Figure 2: Snake topology and interconnection of the Datacenter switch (MLX) with the metro cloud (on the left side), and with the server and the Spirent traffic generator (on the right side).

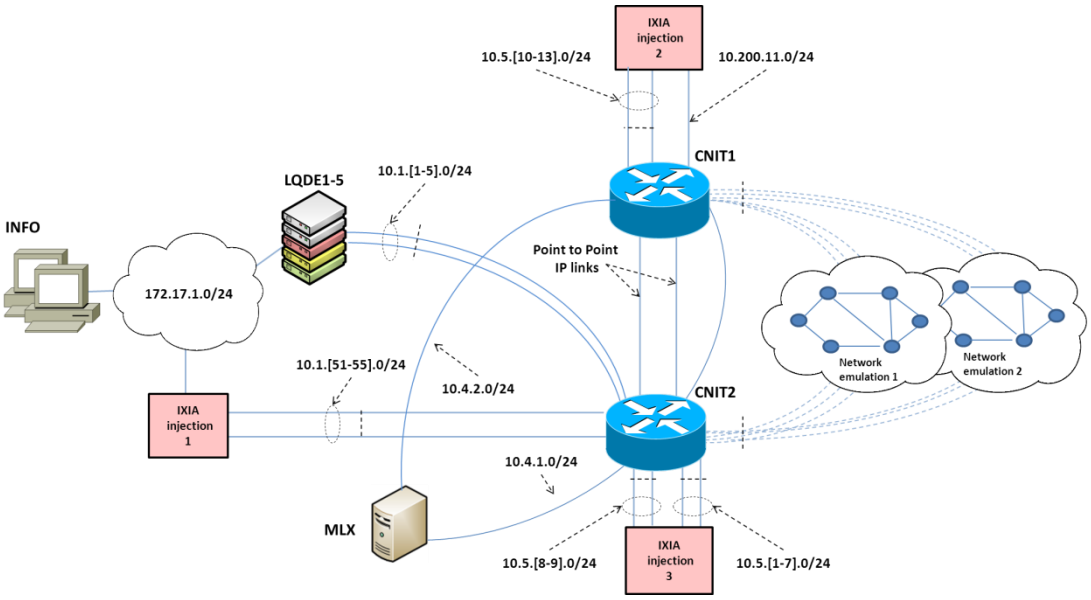


Figure 3: The IP topology of the ECONET demonstration network.

In Figure 3 is depicted the corresponding IP layer topology of the ECONET demonstrator network. In order to evaluate the green routing protocols and algorithms proposed by the ECONET Consortium in the WP5 context, three emulation environments have been set up. These environments are composed by a number of interconnected virtual machines, which were configured to emulate the green control and the management planes of network devices (i.e., routers and transport switches). In other words, the virtual machines run almost the same software implementations of ECONET control and management protocols and algorithms that run on the physical prototypes.

The first emulation environment in the Core Cloud is composed by a topology of 38 routers (36 virtual machines and 2 physical DROP routers), and 77 links (see Figure 4). This topology aims at representing the OSPF backbone area of a medium Telecom operator.

The second emulation environment in the Core Cloud is composed of 6 routers (4 virtual machines and 2 physical DROP routers), and 7 links (see Figure 5). This topology aims at representing an OSPF stub area of a medium Telecom operator.

The emulation network environment in the Transport cloud is composed by 8 switch nodes (6 virtual machines and 2 physical nodes ALU1 and ALU2) and 7 links (see Figure 6). This topology aims at representing a portion of the geographical transport network of a medium Telecom operator. The emulation environment is composed by several instances of Virtual Machines (VM) (see Figure 7), one for each emulated node. The node emulation includes both the control plane local tasks, in charge of path computation and related resource allocations, and the power model of the node, in charge of power consumption estimation used for network optimization. A GAL REST protocol interface is provided to transfer power consumption figures to the central manager, responsible for network power monitoring.

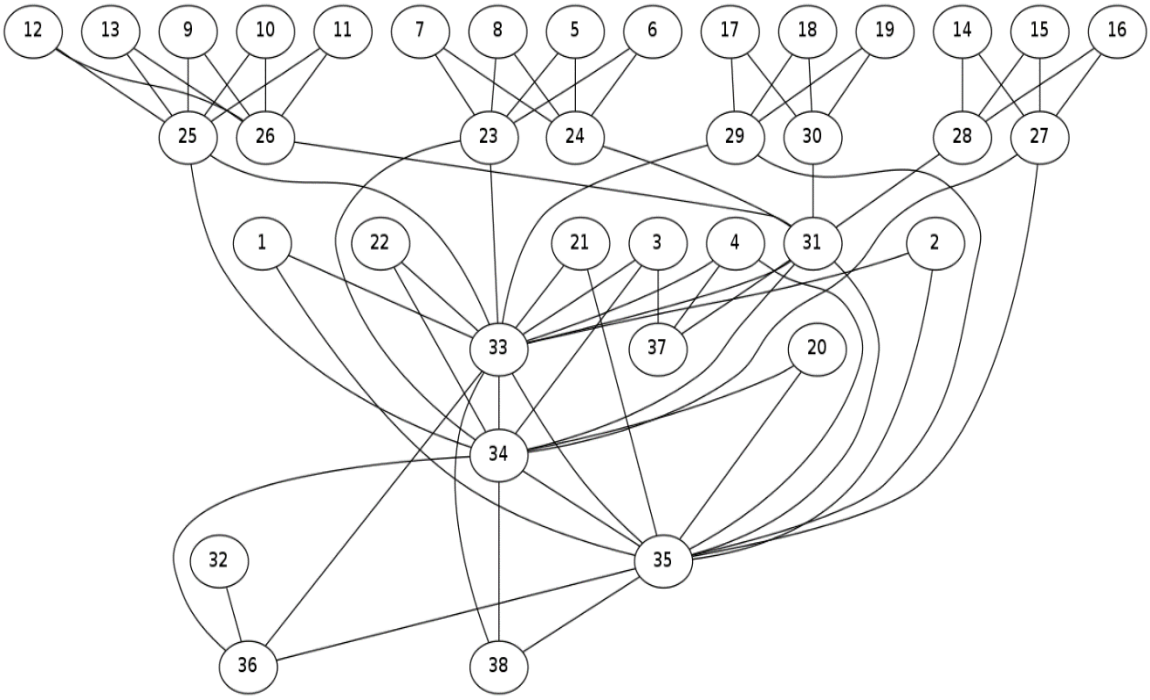


Figure 4: Topology of the Emulation Network 1 in the Core Cloud. Nodes 25 and 26 have been mapped to the physical CNIT1 and CNIT2 devices, respectively.

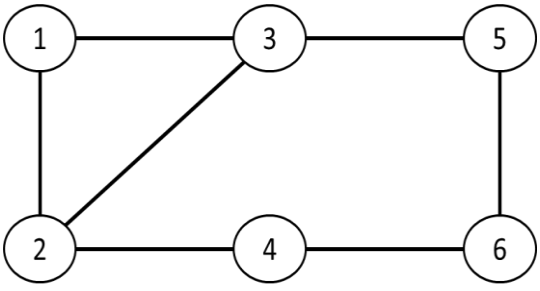


Figure 5: Topology of the Emulation Network 2 in the Core Cloud. Nodes 1 and 6 have been mapped to the physical CNIT1 and CNIT2 devices, respectively.

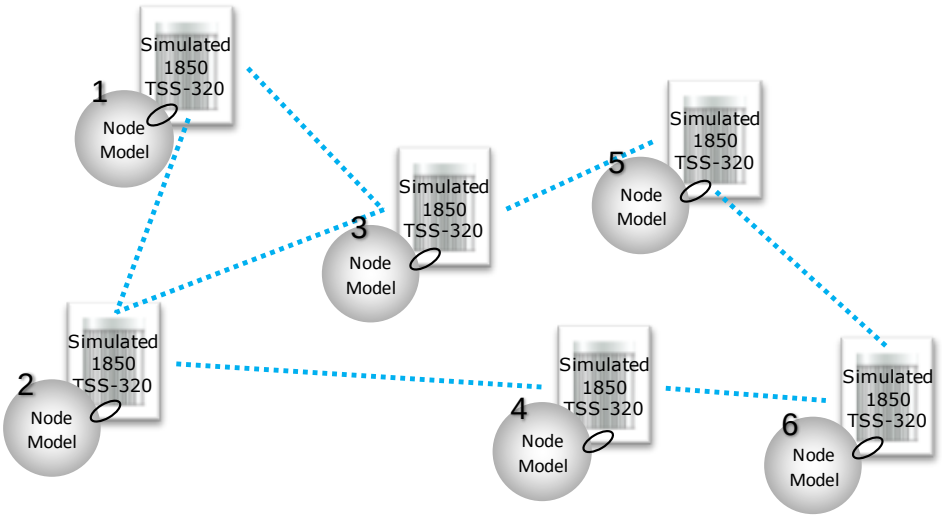


Figure 6: Topology of the Emulation Network in the Transport Cloud.

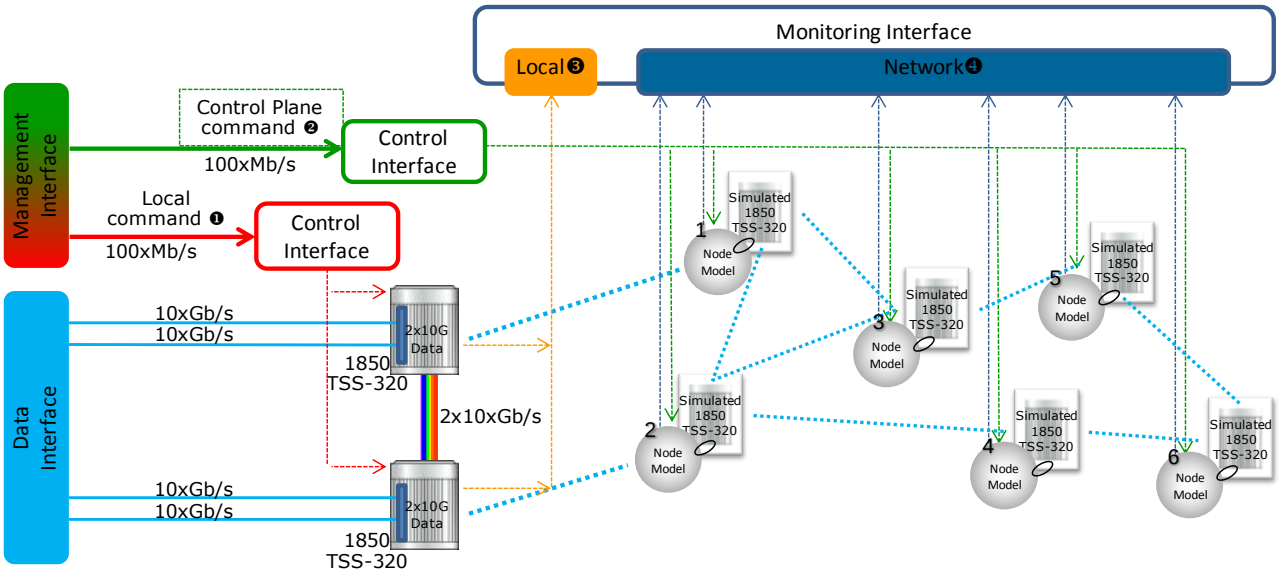


Figure 7: Interconnection of real nodes (1850 TSS320) mapped to ALU1 and ALU2 devices, and emulated network (Nodes 1- 6)

Moreover, the emulation environments have been fully integrated with the physical nodes in the same demonstration cloud, in the sense that the control plane processes of virtual machines exchange data and cooperate with the control plane processes of the physical prototypes. In the router cloud, this integration had been made possible by interconnecting with a number of (Gigabit Ethernet) links the host server where the virtual machines are placed, with the physical prototypes (i.e., the two instances of the DROP router). A different "physical" link has been deployed for each "virtual" link between the physical prototypes and the emulated nodes.

In addition to the physical and emulated demonstration networks, an out-of-band control and service network was set up. This out-of-band network is a simple Ethernet LAN connecting all prototypes/devices mentioned above through their management ports, and the auxiliary control and management systems – like, for instance, the Path Computation Engines for the metro and transport clouds, the control elements of the DROP routers, and the server where the monitoring system was installed.

To provide full connectivity among these devices, the LAN has been configured as a single IP network with address 192.168.0.0/16. A specific value of the third octet of the address has been assigned to each ECONET partner in order to avoid addressing collisions.

4 Traffic generation and profile

The definition of the traffic to be used during the demonstration has been a process highly correlated with the decisions on which prototypes have been selected, which green functionalities were activated, the overall topology of the demonstration network, and the availability and capabilities of traffic generators.

The final result of this process ended with the definition of 7 traffic flows for the physical part of the demonstrator, named F-1, …, F-7. In more detail, F-1 – F-6 are traffic flows with "night & day" profiles that cross various clouds of the demonstration network, while F-7 is only used to generate a bias in the traffic volume inside the Datacenter cloud to correctly trigger local energy saving schemes.

All the flows are generated by means of two traffic generators, which inject and receive traffic to and from the generator from 4 different points, defined as follows:

- *The injection point 1* is attached both to the home networks (and more specifically to the nodes LQDE1-5) by means of 1 Gigabit Ethernet links (1 link x node), and to the ETY2 access switch by means of 4 x 1 Gigabit Ethernet links. All these links physically correspond to the 1 Gigabit Ethernet interface of the Ixia traffic generator, which is interconnected to the ports above by a third Ethernet switch. For isolating traffic going or coming from the different ports to the traffic generator, VLANs have been enabled on the de-multiplexing switch.
- *The injection point 2* is attached to the CNIT1 router, by means of 2 x 10 Gigabit Ethernet links provided by the Ixia traffic generator. These links are labelled in the topology definition as L-21a and L-21b, respectively (see Figure 1).
- *The injection point 3* consists of 7 x 1 Gigabit Ethernet links connected to the CNIT2 router, and provided by the Ixia traffic generator. These links are labelled in the topology definition as L-22a, …, L-22g, respectively (see Figure 1).
- *The injection point 4* is composed by a 10 Gigabit Ethernet link directly connected to the MLX switch in the datacenter cloud. It is realized by the Spirent traffic generator, and used only to generate and receive the traffic of the above-mentioned F-7 flow.

Besides traffic generators (and then additionally to the flows F-1,…, F-7 originated by the injection points 1-4), traffic is also received and transmitted by real hosts attached to the network, and more specifically from:

- PCs in the home networks (the ones of the LQDE4 and LQDE5 home gateways);
- The server in the datacenter cloud.

PCs in the home network are configured to run all the network applications needed to evaluate the Network Connectivity Proxy. The server in the datacenter runs three processes/applications that generate or receive traffic, namely:

- *UDP Traffic Reflector*: it is an application realized by CNIT and MLX aimed at reflecting UDP packets at very high speeds. In details, it runs as multi-threaded application on a desired number of server cores by handling UDP packets destined to the server for a predefined range of layer-4 ports. When a UDP packet is received, the source and destination fields of MAC addresses, IP addresses, and UDP ports are swapped, and the packets sent back to its original source. In order to achieve the maximum possible performance, the UDP traffic reflector has been interfaced with Mellanox Messaging Accelerator (VMA) [1] libraries, which allows an almost direct access of the application to the HW resources of the ConnectX-3 Host Card Adapter (HCA). As discussed later on, the UDP Traffic Reflector application has been used

to reflect the F-3 and F-5 flows back to the injection point 2, and the F-7 back to the injection point 4.

- *Remote Chat Application*: It is an ad-hoc application jointly developed by INFO and CNIT to support the experiments on the Network Connectivity Proxy.
- *TCP Test Server*: It is an already existing application [9], developed in the ECONET context, which aims at generating TCP flows and measuring some performance indexes (e.g., download times, packet losses, etc.).

All the traffic flows previously introduced (F-1, …, F-7) are composed by UDP packets, and in more details include the following fields:

- Layer-2 header: IEEE 802.1 MAC header with (where specified) a single VLAN tag (IEEE 802.1q);
- Layer-3 header: IP header with no extensions;
- Layer-4 header: UDP header;
- Padding: the sizes of the packets follow the Internet Mix (IMIX) distribution.

Where possible, in order to emulate real traffic features, a pool of source/destination IP addresses have been used at traffic generators: this solution allowed to correctly trigger (de-)multiplexing and load balancing mechanisms, which generally work on a per-flow basis, included in the prototypes. Table I reports all the details and parameters used to configure F-1, …, F-6 flows.

Figure 8 and Figure 9 show the paths of the F-1, …, F-6 flows across the physical demonstration network in two configurations: the one at the centre of the day (when all the systems are active and working at the maximum speed), and the one at the centre of the night (when the largest share of energy saving is reached). Between these two extremes, a number of intermediate configurations are adopted.

For the sake of easiness and debugging purposes, but without any loss of generality, all the flows crossing the access, the metro, the data-centre, the core, and the transport clouds have been bound to a different VLAN identifier on each link. The VLAN identifier has been defined as a three-digit number, where the first one represents the flow identifier, and the other ones the link identifier. The VLAN identifiers are obviously popped, swapped, or pushed by the ECONET prototypes.

Figure 8 and Figure 9 also report the maximum offered load of the traffic flows, and the end-points where the traffic is generated and/or received.

It is worth noting that only F-3 and F-5 are bi-directional traffic flows (since they are "reflected" by the server in the datacenter cloud). All the other flows are unidirectional, since (*i*) traffic generators do not allow to easily reflect received traffic, (*ii*) the speeds and the numbers of interfaces of available traffic generators were not enough to support further bi-directional flows, and (*iii*) some of the traffic is directed toward loopback interfaces, and it is discarded as it is reflected.

As reported in Table I, the direction of unidirectional flows were the following:

- F-1 and F-2 are generated from the injection point 3 toward the injection point 1.
- F-4 is generated and received from the injection point 2, and it follows a counterclockwise loop with respect to Figure 8 and Figure 9.
- F-6 is generated from the injection point 3 toward the injection point 2.

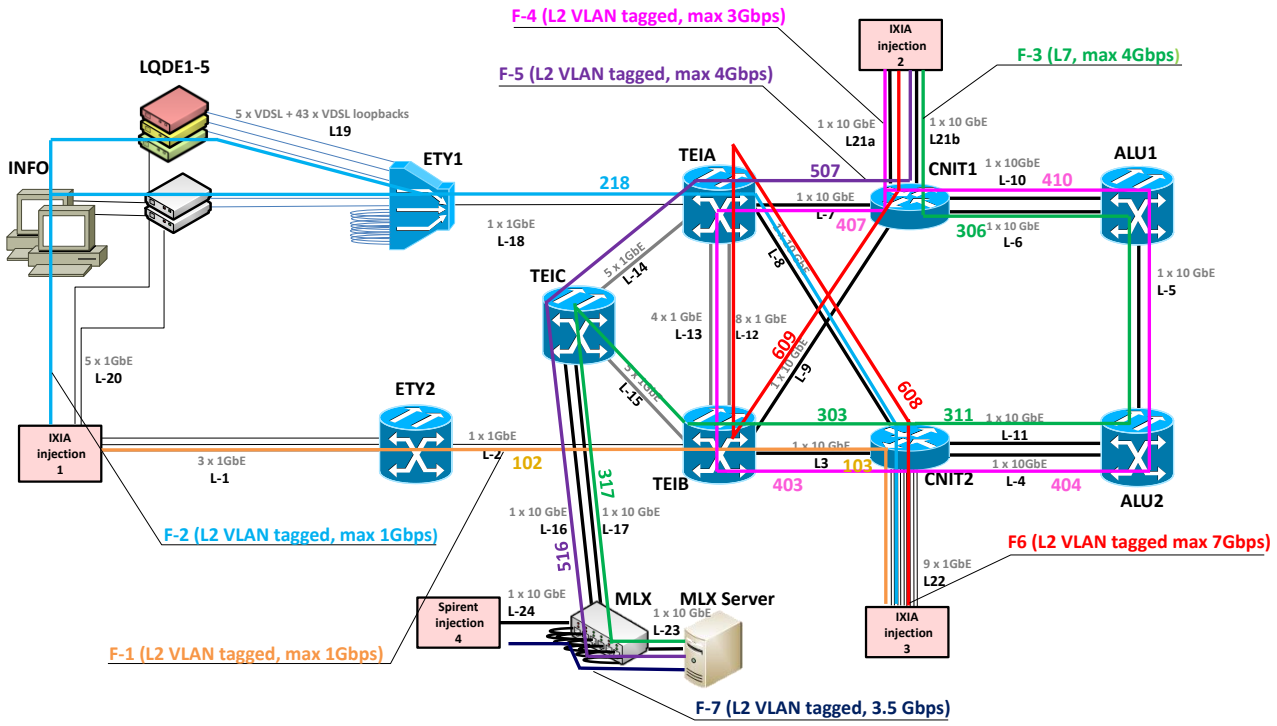


Figure 8: Paths of the traffic flows in the physical demonstration network in the day configuration.

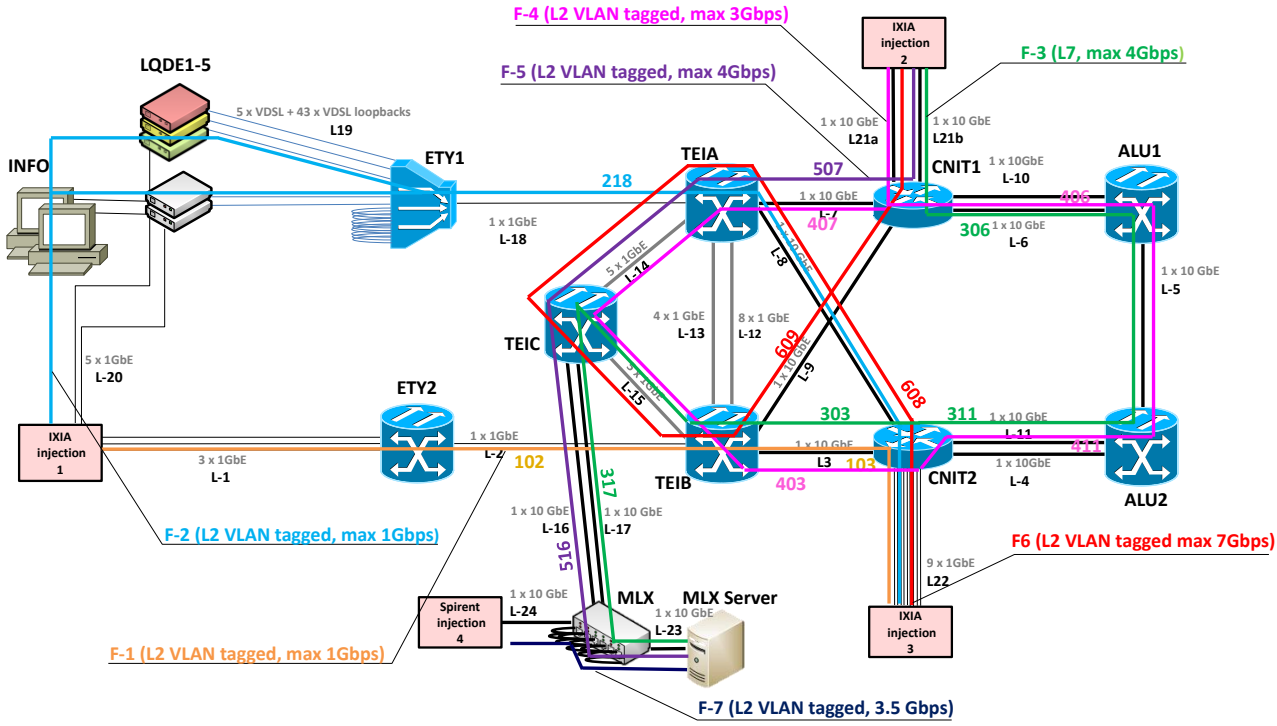


Figure 9: Paths of the traffic flows in the physical demonstration network in the night configuration

Regarding the traffic load fluctuations according to time, the data collected in the measurement campaign performed in ECONET WP2 [11] [12] were exploited. As already done during the DROP demonstration in the 2nd year, the average link load profile coming from the measurement campaign was used as shaping curve for the F1, …, F6 flows.

Given intrinsic problems in generating fluctuating traffic at the traffic generators in an automatic way, it was decided to divide the 24-hours in 300 time slots. Consequently, each time slot represents approximately 5 minutes (or, more precisely, 4 minutes and 48 seconds) of a day. Inside each time slot, the offered load is kept constant.

To speed up demonstration experiments, time slots have been generated at an accelerated time. In detail, it was decided to update the time slots every 20 seconds (thus, corresponding to an acceleration factor of approximately 15x). This time interval has been considered the minimum "safe" length of the time interval that allows the green mechanisms and OAM processes to acquire all the required data, and to fulfil all the needed operations in a slot. Consequently, in each complete run of the demonstration experiments, a daily profile has been reproduced in 1 hour and 40 minutes.

As described later on in this document, the use of accelerated time caused some collateral synchronization issues on those green mechanisms and algorithms that use the time of the day as an input (e.g., a large share of green network control and monitoring mechanisms). The synchronization issues were amplified by the behaviour of the traffic generator, since the operations to update flow traffic loads have been demonstrated to last for a non-deterministic and unpredictable time. To solve these problems, the traffic generator has been programmed to generate a packet with the "emulated" time of the day (that roughly corresponds to the current time slot number), sent in broadcast in the out-of-band control network. All the prototypes requiring this information have been modified to receive this kind of packets from the traffic generator.

The offered loads of F3, …, F6, which somehow represent flows of highly aggregated traffic, are reported in Figure 10, and were simply obtained by multiplying the shaping curve with the maximum offered load for the flow. On the contrary, F1 and F2 represent non-aggregated traffic, since they are terminated inside the home network or at the access level. For this reason, we decided to define them as the superposition of a number of ON-OFF micro-flows, each one directed to a different network termination (e.g., a DSLAM port or an ETY1 switch port). Also in this case, the offered load has been obtained by the shaping curve multiplying by the maximum offered load for that flow, but the output of this operation has been discretized on the basis of the maximum number of micro-flows in the flow (4 for F-1 and 48 for F-2, respectively). Figure 11 reports the final results of these operations.

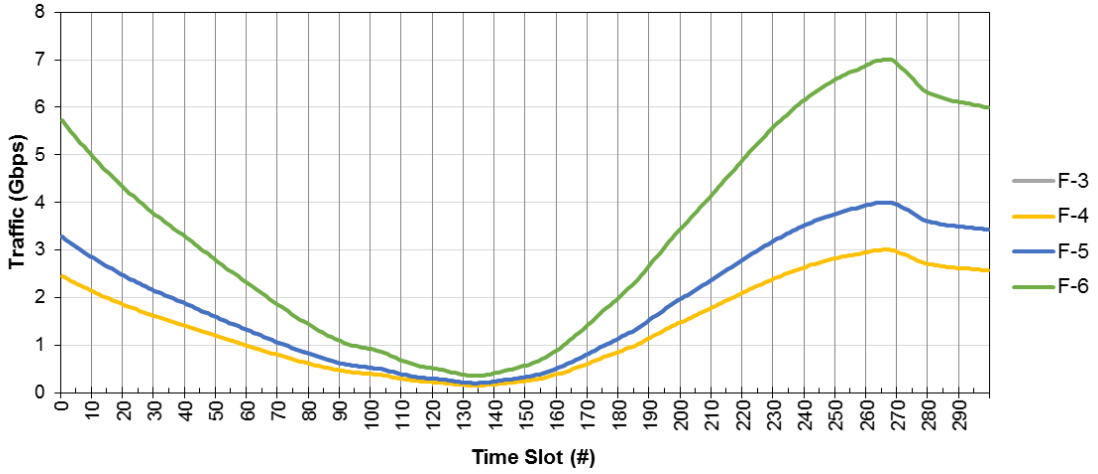


Figure 10: Offered load of the flows F-3, F-4, F-5, and F-6 according to time slots.

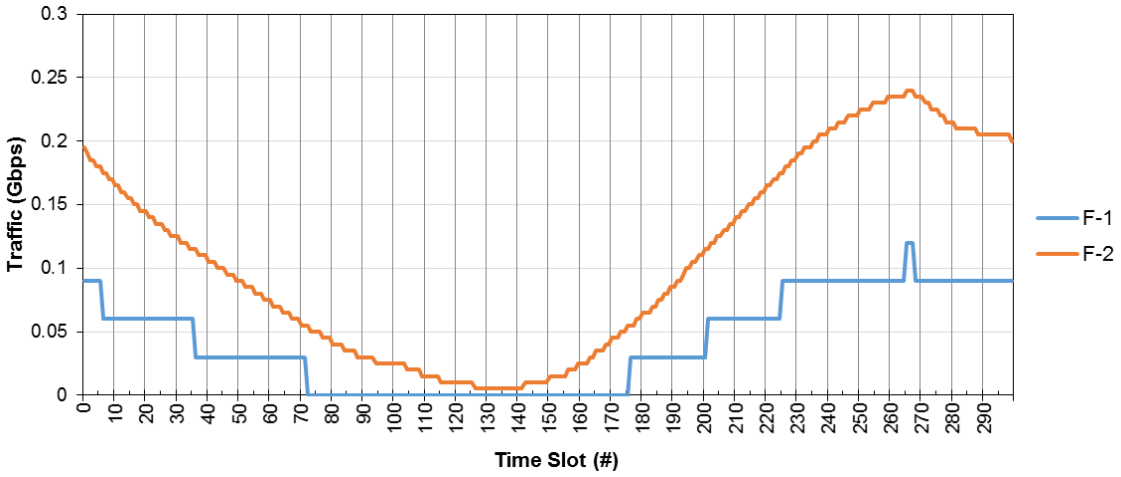


Figure 11: Offered load of the flows F-1 and F-2 according to time slots.

Table 1. Flow parameters and crossed links.

Flow ID	Min rate (Gbps)	Max Rate (Gbps)	Flow Direction and Terminations	Src IP address	Dst IP address	Day path links,VLAN, and hops				Day path links,VLAN, and hops			
						Day Link ID	Day VLAN Tag	Day End-Point 1	Day End Point 2	Night Link ID	Night VLAN Tag	Night End-Point 1	Night End-Point 2
1	0.000	0.120	IXIA-1 -> IXIA-3	10.1.[51-69].2	10.5.8.[2-20]	L-1	701-719	IXIA-1	ETY-2	L-1	701-719	IXIA-1	ETY-2
						L-2	701-719	ETY-2	TEI-3	L-2	701-719	ETY-2	TEI-3
						L-3	701-719	TEI-3	CNIT-2	L-3	701-719	TEI-3	CNIT-2
						L-22	1701-1719	CNIT-2	IXIA-3	L-22	1701-1719	CNIT-2	IXIA-3
2	0.005	0.240	IXIA-1 -> IXIA-3	192.168.1.3 /public: 10.1.[1-19].3	10.5.9.[2-20]	L-20	no	INFO-1	IXIA-1	L-20	no	INFO-1	IXIA-1
						L-19	no	INFO-1	ETY-1	L-19	no	INFO-1	ETY-1
						L-18	801-819	ETY-1	TEI-2	L-18	801-819	ETY-1	TEI-2
						L-8	801-819	TEI-2	CNIT-2	L-8	801-819	TEI-2	CNIT-2
						L-22	1801-1819	CNIT-2	IXIA-3	L-22	1801-1819	CNIT-2	IXIA-3
3	0.198	4.000	IXIA-2 <-> MLX-Server	10.5.10.[2-254]	10.4.1.2	L-21	321	IXIA-2	CNIT-1	L-21	321	IXIA-2	CNIT-1
						L-6	306	CNIT-1	ALU-1	L-6	306	CNIT-1	ALU-1
						L-5	305	ALU-1	ALU-2	L-5	305	ALU-1	ALU-2
						L-11	311	CNIT-2	ALU-2	L-11	311	CNIT-2	ALU-2
						L-3	303	TEI-3	CNIT-2	L-3	303	TEI-3	CNIT-2
						L-15	315	TEI-1	TEI-3	L-15	315	TEI-1	TEI-3
						L-17	317	TEI-1	MLX-1	L-17	317	TEI-1	MLX-1
						L-23	317	MLX	MLX-Server	L-23	317	MLX	MLX-Server

Flow ID	Min rate (Gbps)	Max Rate (Gbps)	Flow Direction and Terminations	Src IP address	Dst IP address	Day path links,VLAN, and hops				Day path links,VLAN, and hops			
						Day Link ID	Day VLAN Tag	Day End-Point 1	Day End Point 2	Night Link ID	Night VLAN Tag	Night End-Point 1	Night End-Point 2
4	0.148	3.000	IXIA-2 -> IXIA-2	10.5.11.[2-254]	10.200.11.[2-254]	L-21	421	IXIA-2	CNIT-1	L-21	423	IXIA-2	CNIT-1
						L-7	407	TEI-2	CNIT-1	L-7	407	TEI-2	CNIT-1
						L-13	413	TEI-2	TEI-3	L-14	414	TEI-1	TEI-2
						L-3	403	TEI-3	CNIT-2	L-15	415	TEI-1	TEI-3
						L-4	404	CNIT-2	ALU-2	L-3	403	TEI-3	CNIT-2
						L-5	405	ALU-1	ALU-2	L-11	411	CNIT-2	ALU-2
						L-10	410	CNIT-1	ALU-1	L-5	405	ALU-1	ALU-2
						-		-	-	L-6	406	CNIT-1	ALU-1
						L-21	421	IXIA-2	CNIT-1	L-21	421	IXIA-2	CNIT-1
5	0.198	4.000	IXIA-2 <-> MLX-Server	10.5.12.[2-254]	10.4.2.2	L-21	521	IXIA-2	CNIT-1	L-21	521	IXIA-2	CNIT-1
						L-7	507	TEI-2	CNIT-1	L-7	507	TEI-2	CNIT-1
						L-14	514	TEI-1	TEI-2	L-14	514	TEI-1	TEI-2
						L-16	516	TEI-1	MLX-1	L-16	516	TEI-1	MLX-1
						L-23	516	MLX	MLX-Server	L-23	516	MLX	MLX-Server
6	0.346	7.000	IXIA-3 -> IXIA-2	10.5.[1-7].2	10.5.13.[2-254]	L-22	622	CNIT-2	IXIA-3	L-22	622	CNIT-2	IXIA-3
						L-8	608	TEI-2	CNIT-2	L-8	608	TEI-2	CNIT-2
						L-12	612	TEI-2	TEI-3	L-14	614	TEI-1	TEI-2
						L-9	609	TEI-3	CNIT-1	L-15	615	TEI-1	TEI-3
						-	-	-	-	L-9	609	TEI-3	CNIT-1
						L-21	621	IXIA-2	CNIT-1	L-21	621	IXIA-2	CNIT-1
7	3.5	3.5	Spirent <-> MLX-Server	10.5.13.1	10.5.13.2	L-24	724	Spirent	MLX	L-24	724	Spirent	MLX
						L-23	723	MLX	MLX-Server	L-23	723	MLX	MLX-Server

Regarding the emulated networks, the following traffic scenarios have been considered.

4.1 Emulation Network 1 (Core Cloud)

The traffic matrix for the emulation network 1 has been composed of 154 flows with a fluctuating offered load according to the time slot. The flows cross the network between the following end-points (with reference to the topology shown in Figure 4):

- From nodes 1-22 to nodes 31-35;
- From nodes 1-22 to node 38;
- From node 33 to nodes 1-22.

Figure 12 reports the aggregated offered load (i.e., the sum of the offered loads of all the 154 flows) of all the aforementioned traffic flows according to the time slot.

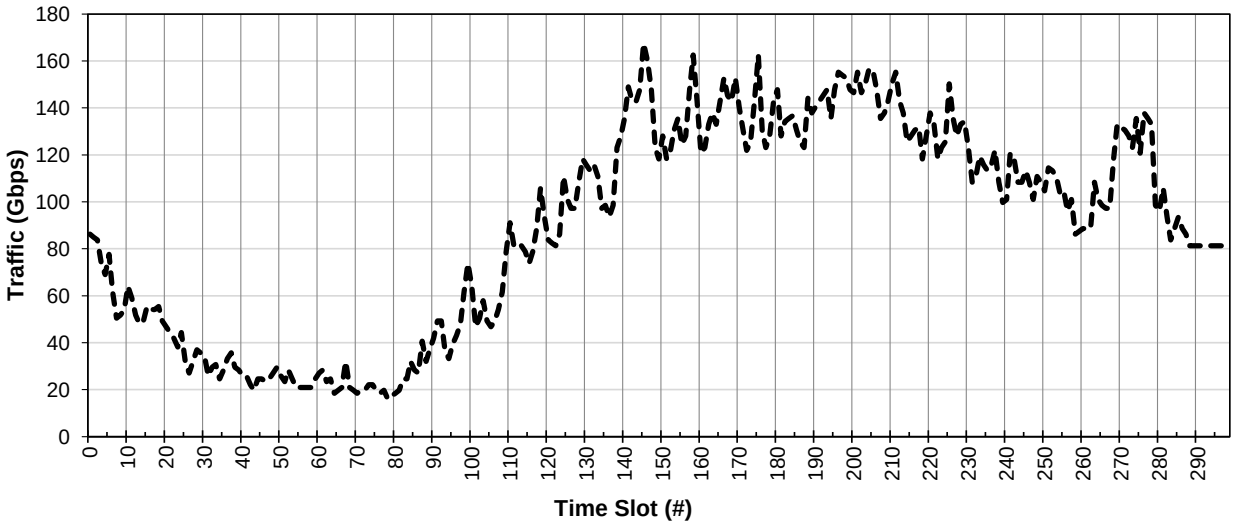


Figure 12: Aggregated offered load according to the time slot for the emulation network 1.

4.2 Emulation Network 2 (Core Cloud)

In the Emulation Network 2 in the core cloud, a single flow has been defined. This flow crosses the network between two additional virtual machines connected to routers 1 and 5 with respect to Figure 5. As in all the other traffic flows, the offered load is fluctuating according to the time slot, and the generated volume is shown in Figure 13.

It can be noted that, in this specific emulation environment, the traffic levels have been kept significantly low (some Mbps), given the scalability issues discussed in Section 7.2.3.

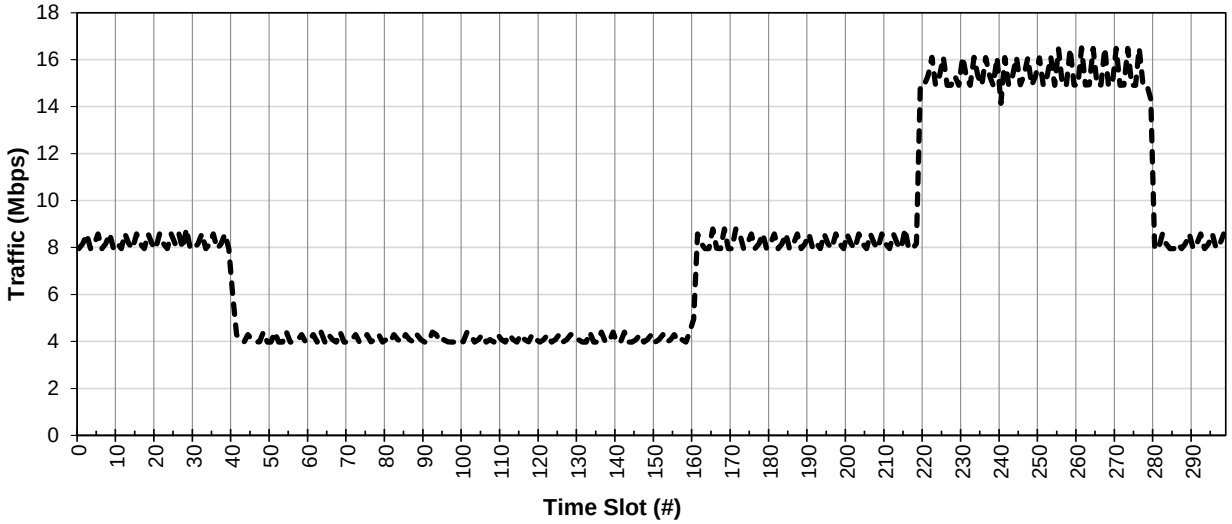


Figure 13: Offered load of the emulated traffic flow crossing the emulation network 2 according to the time slot.

4.3 Emulation Network 3 (Transport Cloud)

In case of the third emulation network, whose topology is reported in Figure 6, a different approach with respect to the previous emulation networks was pursued. Flows F-3 and F-4, passing through the ALU1 and ALU2 physical devices, are considered to follow a more complex path in the virtual network, as L-5 does not exist, and additional (virtual) hops need to be crossed. In detail, the traffic flows are considered to follow the paths depicted in Figure 14. An additional flow is supposed to cross the virtual network between nodes ALUF and ALUD with a reserved bandwidth of 90 Gbps. All the links are supposed to have a capacity of 100 Gbps.

The aggregated (F-3 and F-4) traffic flowing across ALU1 and ALU2 shows a significant day/night load fluctuation. This information is used to trigger the status change of the network nodes, where resources are switched on/off accordingly. More precisely, the interconnection with the CNIT1 and CNIT2 routers is provisioned at 2x10Gbps during the day, while it is reduced to 1x10Gbps during the night-time. This behaviour is synchronized with the control plane of the two neighbouring routers, and also with the Time of Day (TOD) distributed by the Ixia traffic generator over the out-of-band signalling network.

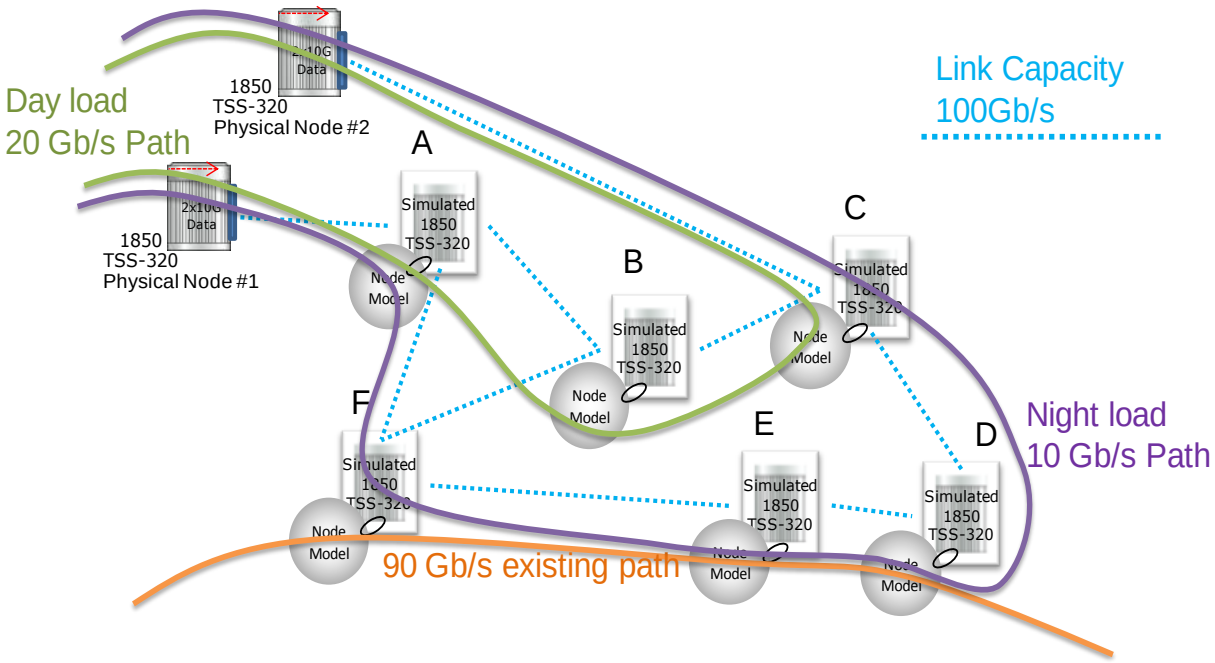


Figure 14: Transport topology (physical and virtual nodes) and emulated flows. The green and the violet lines correspond to the paths of the F-4 and F-3 flows in the transport cloud during day and night configurations, respectively.

5 Functional Architecture of the Demonstrator

The demonstrator includes a significant number of the energy-saving mechanisms and solutions proposed in ECONET Work Packages WP3 and WP5 (summarized in Table 2), as well as different implementations of the Green Abstraction Layer (GAL), which have been integrated in all the physical and emulated prototypes.

Figure 15 provides a first outlook of which kind of green extensions have been included in the demonstrator clouds. In more detail, all the clouds include one or more energy-saving mechanisms at the data-plane, since it is a necessary condition to modulate energy consumption according the traffic load. ECONET solutions at network control plane (i.e., NCPs) are available only at the metro, core and transport networks, where the topologies are characterized by a higher meshing degree, and, consequently, multiple traffic paths are available.

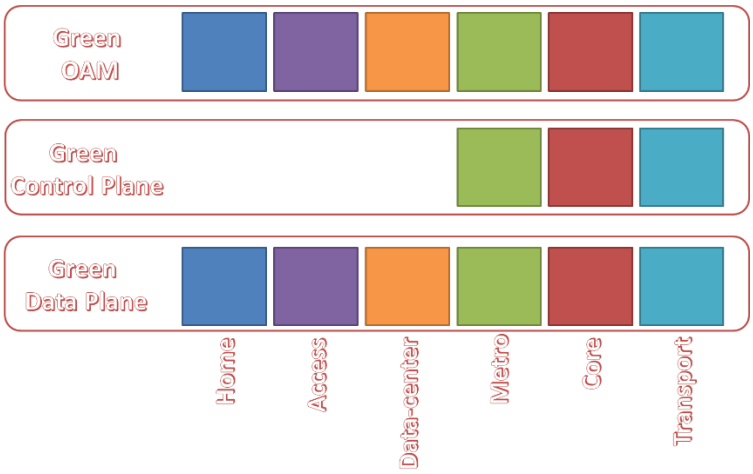


Figure 15: Overview of the green mechanisms available in the demonstration network clouds.

All the clouds have been integrated with a global Green OAM framework, realized in the above-mentioned 7th cloud of the demonstrator, which is able to acquire GAL-like information and usual network performance indexes (e.g., bits and packets sent and received by the devices). The Green OAM framework is physically installed on a server connected to the out-of-band network, and, consequently all this information is sent from the management port of the prototypes.

The GAL-like information is provided to the central framework thanks to both the integration of the GAL on every prototype, and the definition, the implementation, and the integration of the GAL REST protocol [2]. In fact, it is important to recall that the GAL is an API for power-management internal to network devices, and, by default, it cannot be used as network protocol to exchange power management and monitoring information among devices. To carry through the network this kind of information, the ECONET Consortium defined a number of extensions to legacy signalling protocols, which are specific to current architectures and technologies used in the various network segments.

In order to have a single framework monitoring and managing the operations of the whole demonstration network, the ECONET Consortium decided to design a new protocol, the GAL REST, to be integrated in all the network clouds. This new protocol has been designed as a simple mapping of the original GAL data structures and methods on a simple HTTP interface using a REpresentational State Transfer (REST) scheme. Additional details on this protocol can be found in [2].

The resulting protocol is so close to the original GAL specification and the protocol interface so high-level and simple, that it fits well all the heterogeneous technologies and architectures available in the different demonstrator clouds.

In this respect, it is worth mentioning that different approaches have been applied to integrate the GAL (and, consequently, the GAL REST protocol). As made evident by Figure 16, which reports a global overview of the functional architecture of the control- and management-planes included in the demonstrator, these approaches were caused by the different architectures and technological features of the various clouds.

In detail, where NCPs are not present (e.g., in the home and access networks, where the network is not meshed and there is no need for routing protocols), the GAL (and the GAL REST) interface is directly exposed by the device.

In case of meshed networks, the integration approach mainly depends on the type of network control architecture. In case of distributed network control, which is typical in IP networks running the OSPF and BGP routing protocols, the GAL (REST) interface is still exposed by each node independently of the other devices in the cloud.

In case of centralized network control, which is the standard situation in metro, transport and datacenter networks, the most preferred approach has been to integrate the GAL (REST) interface at the network controller, on top of the NCP level. It has to be underlined that, in the datacenter cloud, the MLX reference design includes a GAL (REST) interface both at the network device and at the network controller. However, since it was not feasible to use multiple datacenter switches in the demonstrator for logistic problems, the network controller has not been set up in the demonstrator.

These different approaches, which are feasible thanks to the flexibility of the GAL structure, lead to important general observations on the relationship between the GAL itself and the NCP protocols and algorithms. Integrating the GAL interface on top of the NCP allows exposing not only the power states of the devices (or, more precisely, of their logical resources), but also to expose the states of the network as a whole. This possibility directly comes from the fact that the GAL can acquire all the possible configurations of the entire network from the NCP level, and consequently it can expose them in terms of "aggregated" energy-aware states.

On the contrary, in the case of distributed control, the relationship between the NCP (i.e., the green routing policy) and the OAM framework is not realized in chain as in the previous case. As in current state-of-the-art solutions, NCP and OAM frameworks are independent processes that work almost in parallel. In this case, each single device (router) is responsible for summarizing the directives coming from these two frameworks by means of administrative weights.

It is also important to highlight that, while the ECONET OAM framework retrieves the needed data only from the GAL (REST) interface local to each device, the NCP generally requires additional data sources. For example, in the case of the two emulation networks in the core cloud, the OSPF protocol has been extended to realize green optimization. OSPF, which is a protocol based on the link state scheme, needs to build the entire database of the network in terms of nodes and links. In this case, the data of local resources are retrieved by the standard GAL interface, while the ones related to remote nodes are acquired by means of extended OSPF packets carrying green attributes. The local information (acquired through the GAL) is then propagated to the other nodes by means of the same extended signalling packets.

Table 2. Green mechanisms integrated in the demonstrator clouds and prototypes, and reference to the previous deliverable reports where the mechanisms were introduced.

Cloud	Prototype	Green Mechanisms	Reference
Home Network	*Networked Host*	Network Connectivity Proxy	D3.3 Green Technologies for Network Device Data Plane; section 5.3.3
	Home Gateway	Clock frequency scaling processor & DRAM with voltage scaling	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1.2.3
		On-chip processor offload via accelerator blocks	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1.2.3
		Selective Power Off of interfaces	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1.2.3
		VDSL low power mode	D3.3 Green Technologies for Network Device Data Plane; section 5.2.1 1
		DECT, FXS, FXO enable/disable	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1.2.3
		WLAN MIMO scaling	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1.2.3
		Energy Efficient Ethernet EEE	D3.3 Green Technologies for Network Device Data Plane; section 5.2.1.1
Access Network	*DSLAM*	Cut Through Mode with DDR disabled	D3.3 Green Technologies for Network Device Data Plane; section 5.3.2.1
		DSL Line Shut Down	D3.3 Green Technologies for Network Device Data Plane; section 5.2.1.1
	CPE Switch	Cut Through Mode with DDR disabled	D3.3 Green Technologies for Network Device Data Plane; section 5.3.2.1
		TX direction Link Shut Down	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1
		RGMII Ports Shut Down	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1
Metro Network	*Metro Switch*	Reset Partitioning	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.2.1
		Clock and Voltage Partitioning and Scaling	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.2.1
		Selective Power Off	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.2.2
		Energy Aware Routing	D5.3 - Green Strategies at the Control Plane; section 4.3.1
		Local Control Policy	D3.3 Green Technologies for Network Device Data Plane; section 5.3.1
		Network Control Policy	D3.3 Green Technologies for Network Device Data Plane; section 5.2 D5.3 - Green Strategies at the Control Plane; section 4.2
Datacenter Network	*Datacenter Switch*	Power Consumption Measurement	D3.5 - Design of Energy Measurements Technologies; sections 4.3, 5, 6 D3.3 - Green Technologies for Network Device Data Plane; section 6.2
		Voltage scaling	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1.2.2
		Frequency scaling	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1.2.2
		HW shut down	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1.2.2
		Power-Aware PHY optimizations	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.1.2.3
		Heat dissipation and Fan control	D3.3 Green Technologies for Network Device Data Plane; section 5.1.1.2.3
		Width reduction Power Saving	D3.3 Green Technologies for Network Device Data Plane; section 5.2.1.2
Core Network	*Distributed Router*	Energy Aware Load Balancer for Multiple Parallel Pipelines with DVFS	D5.1 - Preliminary Definition of Green Strategies at Control Plane; section 3.1.2.1
		Chip-level Traffic Redirection with Low Power Idle	D6.2 – Final Integration of Device Prototypes with Energy Aware Capabilities and Local Optimization; section 3.1.2
		System Level Taffic Redirection	D3.3 - Green Technologies for Network Device Data Plane; section 5.1.2.2.2
	Distributed Router & Emulation environments	GRiDA routing algorithm	D5.3 – "Green strategies at the control plane"; section 4.3.3
		ESIR routing algorithm	D5.3 – "Green strategies at the control plane"; section 4.3.5
Transport Network	*Transport Switch*	SW/FW and HW support for power ON/OFF commands	D3.3 - Green Technologies for Network Device Data Plane; section 5.1.1.2.2
		Power-down capabilities for optical client interfaces	D3.3 - Green Technologies for Network Device Data Plane; section 5.1.1.2.2
		frequency scaling	D3.3 - Green Technologies for Network Device Data Plane; section 5.1.1.2.2
		HW probes for battery power measurement	D3.3 - Green Technologies for Network Device Data Plane; section 5.1.1.2.2
		HW probes for internal current and voltage monitoring	D3.3 - Green Technologies for Network Device Data Plane; section 5.1.1.2.2

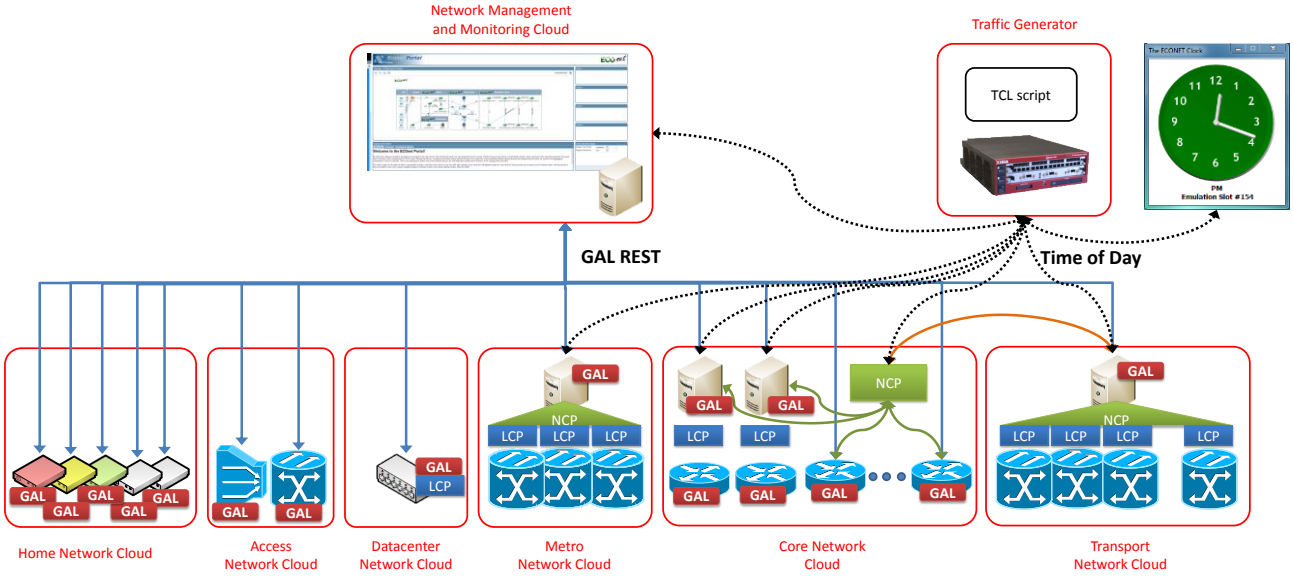


Figure 16: Overview of the functional control- and management-plane architecture of the demonstrator.

A further key integration activity, carried out during the demonstration design and integration phases, was to interface NCPs acting at different layers, and more precisely the one at the IP layer with the NCP at layer 2.5 of the transport cloud. This integration has made visible in Figure 16 by means of the orange line connecting the core and transport clouds.

In more detail, the CNIT1 and CNIT2 routers have been interconnected using two light-paths in the transport network, which are exposed as two direct links connecting the two nodes in the IP overlay network. The transport NCP is able to expose dynamic weights for the two light-paths to the IP layer, so that the router can decide which of the two links is more convenient for sending traffic to the same destination.

Finally, Figure 16 also reports the propagation of the "time of day" message sent by the traffic generator (see Section 4 for additional details on this aspect). In more detail, this kind of message is received by all the devices that need to perform resource provisioning based on this information, or to collect measurements from the demonstrator.

In the former case, we can find NCPs and a large share of LCPs acting at the metro, core and transport clouds. In the latter case, we can find the OAM framework and additional GUI tools that are used to visualize the behaviour of the demonstrator in real-time. The "ECONET clock" application is an example of these tools: it is a simple graphical analogue clock that awaits messages from the traffic generator, and visualizes the time derived therefrom.

6 Parameters collected and Measurements performed

The full-scale benchmarking and validation system, as presented in this document is expected to give evidence that the energy conservation mechanisms are indeed applicable for larger-scale deployments. However, to present these evidences in an easy-to-comprehend way is not a simple task, as the efficiency improvement measures typically involve some coordinated action on multiple devices. Moreover, also the power saving compared to the BAU scenario is required to be computed, and presented at the system level, rather than at the level of the individual devices or components.

Clearly, some functionality is required to present this high-level information, but also providing access to the detailed data underlying the total figures.

The ECONET demonstration architecture thus includes a high-level monitoring/management system, which is discussed in this Section. This system has been developed with two parallel goals in mind:

- To provide a visualization system for the ECONET benchmarking and validation
- To provide a Monitoring/Management framework as a prototype for the supervision of the ECONET energy saving mechanisms; most notably, a system that predominantly uses the GAL Monitoring functions to collect information and present it in high-level overviews.

The following subsections

- Describe the monitoring and management system in more detail;
- Describe the interface used for the bidirectional communication between the manager and the managed entities;
- Describe the basic set of metrics, whose support is suggested for all nodes with the GAL functionality.

6.1 Metrics used by the monitoring interface

In order to make measurements by several ECONET partners comparable, we defined the following basic metric set, to be used by all partners that provide prototypes for the demo.

Note that only a part of the metrics fit into the scope of the GAL [3] (Table 3), while other ones (Table 4) are collected through the mechanism of state-metric data acquisition, described below at the end of this section.

Table 3. Metrics defined in the GAL

Name	*Unit*
Power state	(enum/integer)
Instantaneous power	W
Accumulated power consumption	kWh
Current	A
Voltage	V

Table 4. Metrics defined outside the GAL

Name	*Unit*
Actual bitrate	Mbps
Byte count (in/out)	Mbytes
Packet count	pkts
Dropped count	pkts
Error counts	num

6.2 The ECONET Management & Monitoring System (MMS)

The Measurement and Monitoring system is a principal component of the ECONET management and validation framework, with the following functions:

- Visualization of the topology of the management system, also covering the hierarchy of nodes, modules, components, etc., reflecting the hierarchical structure of the system (see Figure 17).
- Collection of state information (e.g., power mode, service performance and electrical consumption values), and its visualization on the map itself.
- Storage of all collected data in a database, thus making it possible to display historical data or run data analysis in retrospection.

Figure 17: The GUI of the ECONET Monitoring and Management System.

From the users' perspective, the MMS offers a modern and intuitive web interface (the "ECONET Portal') with the following basic workflow steps supported:

- After authenticating into the system, open the top-level overview topology of the managed system (i.e., the ECONET clouds, in this case)
- Clicking on any container or icon in the maps will allow browsing the hierarchy of device groups, devices, modules or components, along with the connections between different entities.
- Each icon will provide information on the states of the corresponding entities. They can either be on the main icons themselves (like icon background indication of power state, or connection line width indicating throughput), or can be represented with small icons displayed on top of the main icons (like device consumption printed in small numbers).
- All the above visualization mechanisms are user-configurable through settings at the bottom of the screen.
- The context menu of entity icons allows selecting any of the metrics defined for the entity to display them in the historical charts on the right side.
- In addition, if the device offers operations to the MMS (like changing the power mode manually) the controls for these operations are also offered in the device context menu.

A main feature of the MMS is that it is automatically configured from data provided by the entities. For instance, each prototype cloud offers topology description and state information through the ECONET GAL (or some other, established mechanism, like SNMP) and the MMS will draw the right topological maps based on this information.

The purpose of the current section is to describe the registration and communication mechanisms applied in the validation system.

First, we describe the XML format used by the topology chart engine on the ECONET dashboard. The following pages summarize the concept of the topology and of the XML descriptors, and specify the valid XML tags, which can be used for the description of the topology, while Appendix A contains an example XML.

Next, we describe the HTTP REST API binding of the GAL, which is the basic mechanism to communicate information between managed entities and managers.

The final part of this section describes an alternative technology denoted as the state metrics, which is used to communicate information not contained in the GAL.

6.3 Function of the topology interface

The Topology interface is one of the three interfaces used between the ECONET Visualizer and the managed ECONET clouds. It is the basic interface that enables the access to further sub-interfaces by specifying the devices, the components and the connections within the managed clouds (see Figure 18).

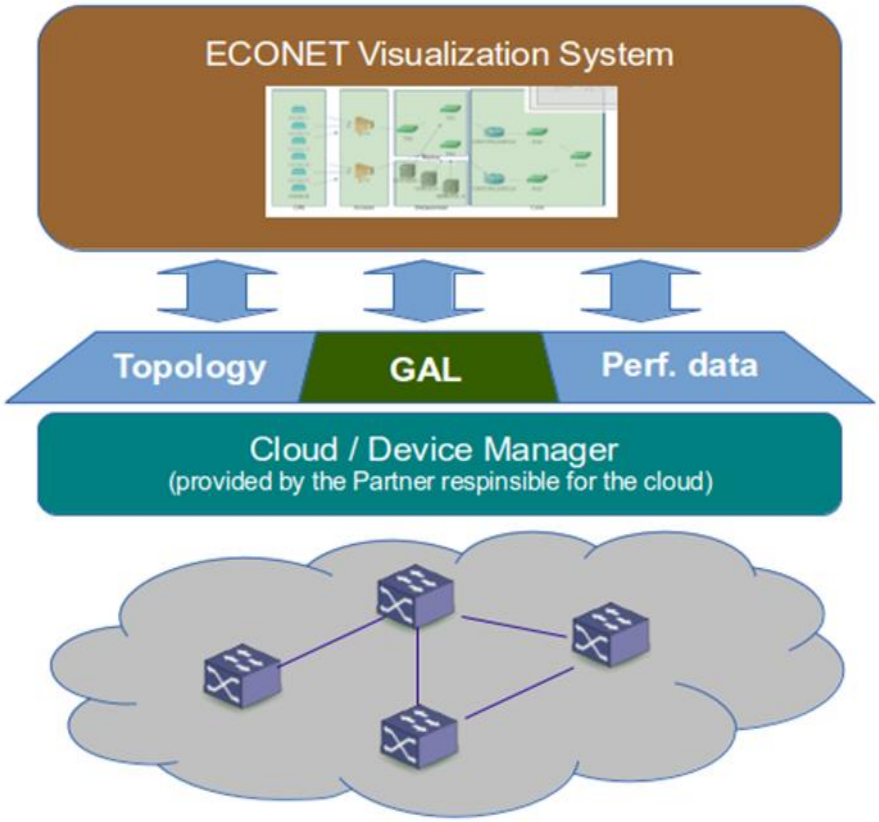


Figure 18: The three interfaces connecting the ECONET subsystems to the central management.

To summarize the other two interfaces (which are specified in detail in [2]):

- The GAL interface is the main ECONET structure used to communicate energy-related information – energy states and energy usage data- between the manager and the managed systems.
- The Performance interface collects status information on the entities that fall outside the GAL scope, e.g., instantaneous interface data rates, processor loads, etc. This interface strongly depends on the Topology interface, in the sense that performance measurements, which point to the managed entities, are specified through the Topology interface, i.e. within the topology description documents (see state metrics in Section 6.8). This is not the case for the GAL metrics, as those are typically defined through the GAL discovery mechanism.

The interface specifies an XML format. The Cloud owners will produce parts of the whole topology by publishing XML documents of their sub-trees; these documents are periodically requested by the management apps and conceptually concatenated into a full tree of the ECONET-managed entities.

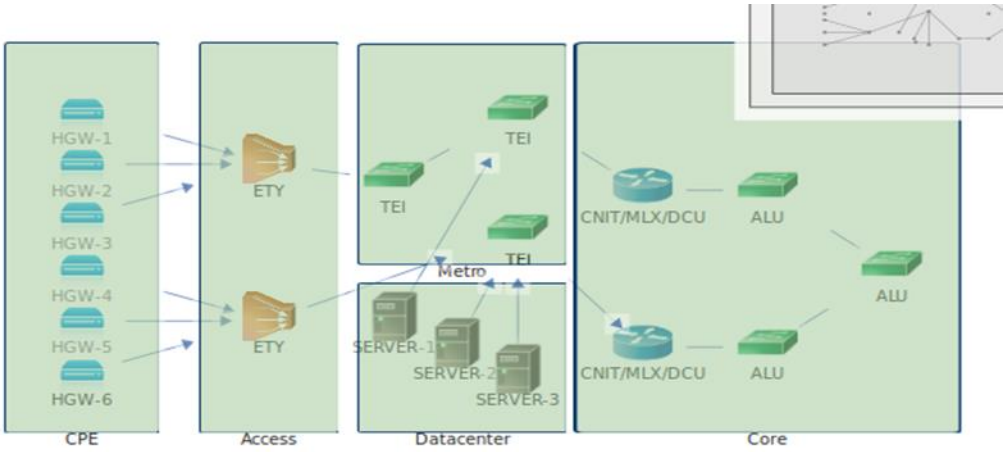


Figure 19: Draft visualization of the "ECONET Universe", with five clouds, devices, and their connections displayed.

6.4 Entities of the topology

The topology is a hierarchy specified by 5 principal layers (see Figure 18):

- **Universe**: the single root of the full system (i.e., the entire ECONET demonstrator setup). This is managed by the dashboard administrator.
- **Cloud**: a partition within the universe containing devices, which belong together through type similarity or administrative authority.
- **Device**: represents a physical device in the demonstration network.
- **Module**: a card, chassis, or other replaceable module within the device.
- **Component**: the smallest entity, which can be characterized by performance/load and possibly also by (measured or calculated) energy usage. Examples are ports, CPUs, fans, displays, etc.

Table 5. Hierarchy of valid containers and contained entities.

Child / Container	Cloud	Device	Module	Component
Universe	√			
Cloud		√		
Device			√	√
Module			√	√
Component				

The XML definition also supports the concept of **Connections** between components. The valid containment relationship for the topology is specified by the matrix in Table 5.

The topology is specified in XML format, which may be a single XML document describing the whole universe, but more typically a set of XML documents with a hierarchical reference relationship. In other words, the root XML only describes the Universe and possibly some of the Clouds including their internals, while the contents of other Clouds may be described in referenced XML documents. Those documents may again provide only a partial description, referencing other XMLs to describe some details of some devices, modules or components.

A possible example for contained XML documents is shown in Figure 20.

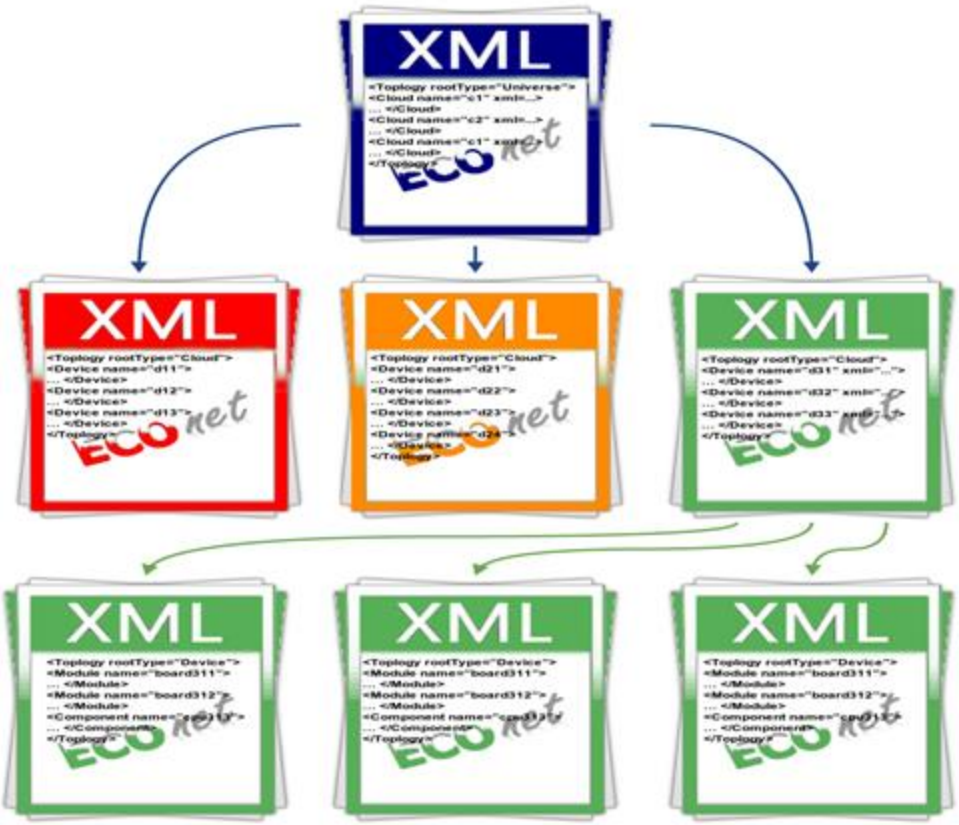


Figure 20: Example hierarchy of XML docs describing the topology Universe. Documents of different colours are produced by different partners. The blue XML describes the 'Universe' and is maintained as part of the central management system.

6.5 XML Structure

The root tag is `<Topology>`. This tag contains the description of exactly one entity, whose type is specified through the `rootType` attribute, which can be any of `'Universe'`, `'Cloud'`, `'Device'`, `'Module'` or `'Component'`. The attribute `'top_id'` must be set for all Topology descriptors, except for the one describing the `'Universe'`, and must be equal to the `top_id` used in the referencing tag of the parent XML.

6.5.1 Contained entities

Each file may contain `<Cloud>`, `<Device>`, `<Module>` or `<Component>` tags compliant to the containment rules shown in Table 5 above.

Entities are identified by ID attributes assigned and interpreted through the following rules:

- A locally unique `id` attribute may be assigned.
- A globally unique (i.e., unique for the full universe) `top_id` may be assigned. This is mandatory if the `id` attribute is unset, otherwise it defaults to `'<parent_top_id>-<id>'`, where *`parent_top_id`* is the top_id of the root element of the containing document.
- A globally unique `gal_id` may also be assigned. If missing, this defaults to the explicit or defaulted `top_id` of the enity.

Further typical attributes and tags are:

- `<Type>`: a specification for the function/kind of the entity; e.g., meaningful types for Devices are “Router”, “Switch”, “Server”, “MSAN”, etc. Among other things, the `type` attribute selects the default icon used for the entity, and other type-specific behaviours that might also be possible in the future.
- `<Name>`: The name of the entity used in diagrams (instead if the top_id, which is the default).
- `<Description>`: A text of up to 64 characters shown in the details panel of the entity.
- `<Gal_URL>`: If there is a GAL/REST interface at this level available for the monitoring/visualization systems, the URL of this interface is specified in this tag.

6.6 Visualization

There is also an optional tag named `<Visualization>`, defined in the content of any entity tag. As the name implies, it only affects the way an entity is displayed on the topology charts.

If this tag is missing (or just partially specified), then the dashboard automatically selects an icon and position and visualize it on the topology charts.

- `<Icon>`: the name of the icon used to display the entity without its internal structure. The default for this value is an Icon calculated from the tagname (Cloud, Device, etc.) and the type of the entity. This can be a resource name on the management server (with '.png' as the default extension) or a URL to an external resource.
- `<Container_Icon>`: the name of the icon used to display the entity when its internal structure is also revealed. The default for this value is a white box with the entity's effective 'Icon' displayed in the header of the box (see Figure 20). This can be a resource name on the management server (with '.png' as the default extension) or a URL to an external resource. As Components have no internal detail, this attribute has no effect for those entities.

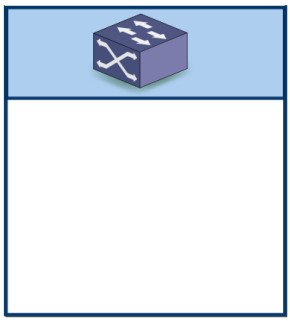


Figure 21: Default container icon for a switch device.

- `<Position x=... y=...>`: specify a hint on the icon's position relative to the parent entity when both entities are shown, or when the parent is the root of the topology shown. As general rule, the center of a child icon is specified relative to the center of its parent icon; "`x=0 y=0`" is the very center of the parent, `x=-100 y=100` is the top-leftcorner, etc.

6.6.1 Connections

Connections can only be defined between Components. They make sense only for certain types (i.e. ports), but there is no rule to limit the connections between components. The connections are bidirectional by default, but with the "direction" attribute, the default can be overridden. The parameters are specified as:

- *UNIDIRECTIONAL* 0
- *DIRECTIONAL* 1
- *BIDIRECTIONAL* 2

Ports are specified through the `<Connection>` tag that has to be embedded in either the `<Component>` tags of connection end-points. The other component is specified through its `top_id` assigned to the `peer` attribute.

Connections themselves may also have a `top_id` attribute. If a connection is defined in both Components and the top_id-s are equal (or missing on at least one side), these definitions are considered to be redundant and only a single Connection is considered.

Connections may also have <Visualization> tags, with the following sub-tags:

- `<Width>`: Relative stroke width in the 0.1-10 range, 1 is the default.
- `<Style>`: `'normal'`, `'dashed'` or `'double'`.
- `<Colour>`: any of the colour specifications valid in HTML; e.g. "`red`", "`ligthblue`", "`#A53214`".

6.7 GAL rest API – an overview

(Please see D5.5. for a complete definition of this interface.)

REST (Representational State Transfer), is a popular network API architectural style, which is characterized by the consistent usage of the limited HTTP command vocabulary (GET, PUT, POST, DELETE, etc.), to implement functionality through a series of read and write operations on an object-oriented database. Sections of this database are represented as documents (XML documents in our case) during the client-server communication.

The basis of REST is typically a conceptual hierarchical "document", which describes the entirety of the information covered and handled by the API. We call it in our case the "GAL REST XML tree" which describes all information covered by the GAL (including writeable state variables). Since the GAL is hierarchical, this suits the tree concept very well: each level represents a GAL resource, with children corresponding to the subordinate resources (e.g., devices in a network, or ports on a line-card, etc.).

The GAL REST API works by providing access to parts of this tree through HTTP operations. For example,

- *discovery* is an HTTP GET query where the URL path and the URL query parameters specify the information to be returned;
- *provisioning* is an HTTP PUT, with the state specified as an XML fragment in the document part of the HTTP request;
- *commit* and *rollback* are POST operations.

In all cases the information returned consists of an error code and/or a document containing the data requested; e.g., for successful operations HTTP return code 200/OK is returned, while operations on nonexistent resources return 404/NOTFOUND, and invalid operations return 401/BAD REQUEST status codes.

6.8 State metrics

In addition to static (i.e., rarely changing) topological data described in this document, and in addition to energy usage status represented by the GAL interface, networks may have additional status (i.e., frequently changing) properties which may be relevant for management, and for visualization, but -being unrelated to energy usage and power modes- they are not covered by the GAL. Examples for such states are interface speeds (bitrate, packet rate) or CPU loads, memory usage, ambient temperature, etc.

While the actual values of these states are collected by the monitoring system through the "Performance monitoring" interface, the XML topology descriptions are in charge of the specification of these data sources. This is made possible through optional `<StateMetric>` tags on any entities.

This `<StateMetric>` tag supports the following attributes.

- `name`: name of the metric
- `type`: `'numeric'`, `'numeric-accumulated'` or `'text'`, specifying whether the state is basically a number, or if it is just some text, like `'off'`, `'operational'` or `'booting'`. Default is `'text'`. The difference between numeric and numeric accumulated values is that `'numeric'` is a momentary value (like speed), while `'numeric-accumulated'` refers to a value that accumulates over time (e.g., bytes transferred).

- `units`: the measurement unit needed to interpret numeric values (applies for format=`numeric` only.
- `range`: the valid range for numeric metrics (e.g. "`0.0 .. 20.0`", or the valid values of text-format status indicators (e.g. "`off|operational|booting`").
- `access`: the method to be used by the performance data collection system to access the current status of this status parameter; it can be any specification conforming with the following formats
  - "`http://<any url>`", "`https://<any url>`", "`ftp://<any url>`" – data is to be collected by accessing the http/https/ftp URL specified. Accessing this URL should return a single number or a string, depending on the type of the metric.
  - "`snmp://<ip_address>[@<community id>]/<oid>`": the value is accessed through an SNMP query.
  - "`ssh://[<user>@]<ip_address>:<command_with params>`" : the value is the output of a command executed by ssh.; i.e., the output is a single number or a string, depending on the type of the metric.

7 Demonstrator Cloud Layout

7.1 Optical Transport Cloud

The optical transport network is based on ALU-I Optics products. The choice to base the demonstration activity on 1850 TSS320 was justified by the fact that this product is developed by the Italian team, able to exploit ECONET research by adapting commercial products to demonstrate the effectiveness of the ECONET approach. However, the same feature could be easily imported by other products belonging to the family, such as PSSxx.

As explained in Section 4 (in the "Emulation Network 3" subsection), this cloud has been partially based on virtual emulation of a larger scale transport network. In detail, according to the optimization process architecture introduced on D5.5, Section 3.2 [2], an off-line optimization procedure has been implemented based on the NASK optimization algorithm. The algorithm provides information about the list of links used to implement the requested path, minimizing the power figure, while preserving the requested QoS.

Figure 22: Alcatel-Lucent 1850 TSS320 PTN Node.

7.1.1 Function of the device in the network

The device used for the demonstrator can be classified as a Packet Transport Network (PTN) device, being able to consolidate traffic in the Metro-Core area and efficiently transport it across the optical backbone. This device uses MPLS-TP technology to forward the traffic across the network with sophisticated capability of implementing and differentiating QoS, according to the traffic requirements. The power saving opportunity in this network area is not so large for two main reasons:

a. The number of devices is not very large (very far from the multiplicity of access devices)

b. Dealing with consolidated traffic the network design tends to use as much as possible the implemented pipes.

Nevertheless, by using ECONET technology, significant saving has been demonstrated.

7.1.2 Type of technology used

According to the identified scenarios, the saving opportunities can be and have been achieved by applying the following criteria:

1. Technological choices and autonomic behaviour of the node with respect to the power consumption. This allows saving an important percentage of the power consumption, without impacting the actual management/configuration of the network. As reported, this allows 59% saving, going from Rel. 3 to Rel. 7 of the product.
2. Long-term traffic variations can use the capability of allocating the traffic path by optimizing the network power with the choice of a smart strategy to switch on/off the unused resources. In the demonstration use-case, we experimented a saving in the range of 15-20%.

Furthermore, a power-monitoring functionality has been implemented both in real and in simulated nodes. It is based on the GAL technology and implements GAL REST interfaces, being able to talk with the centralized power-monitoring framework provided NVR.

7.1.3 Characteristics of the device with respect to both transmission and energy aspects

The HW prototype used in the demonstrator shows the behaviour of Figure 23 for different power modes.

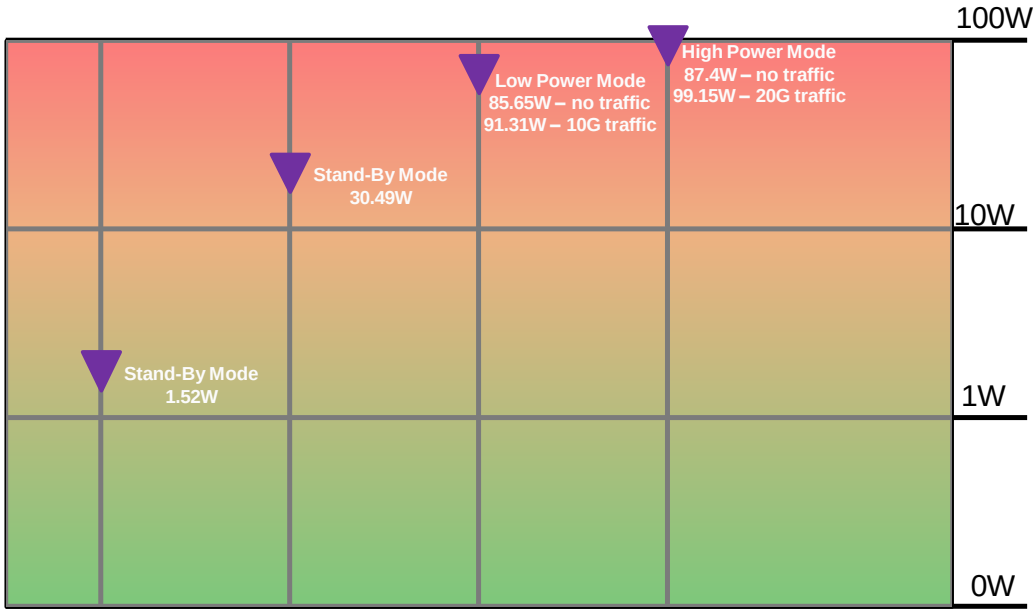


Figure 23: Power behaviour of the 2x10Gbps Data Board.

Similarly, the power behaviour with respect to the traffic load is represented by Figure 24. In this case, we observe a power variation of about 10W ranging from 0% to 100% of traffic load, corresponding to about 10% of the total power consumption.

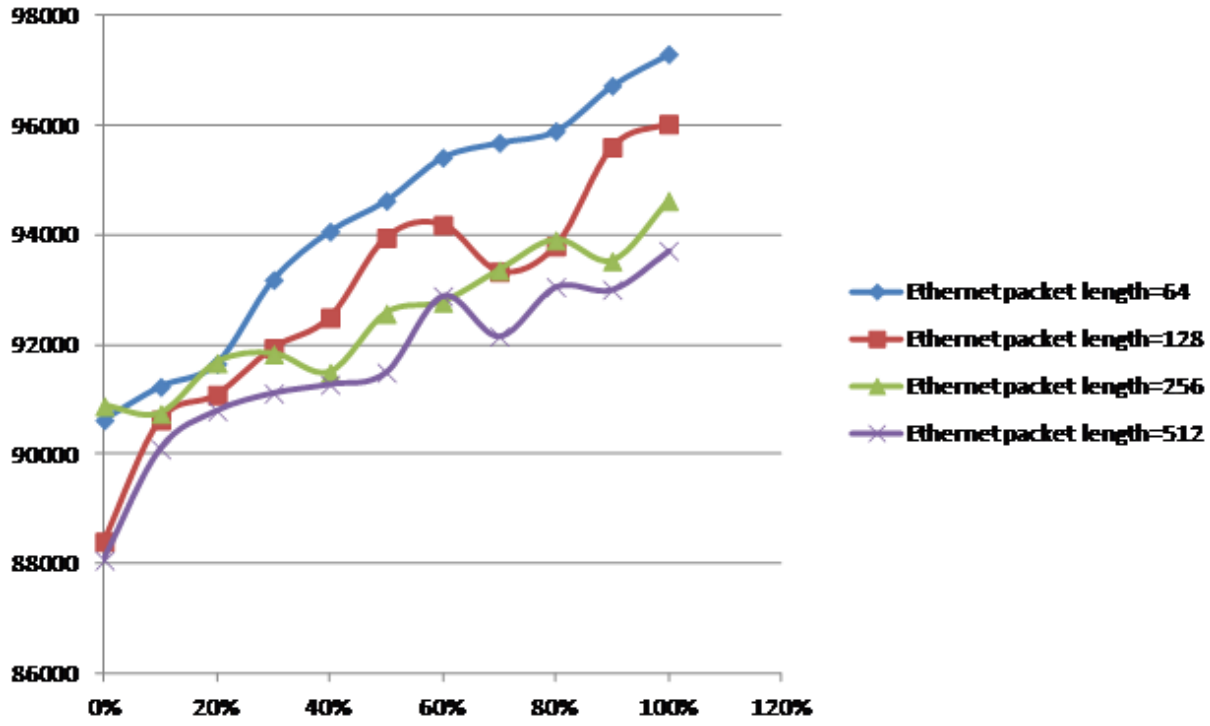


Figure 24: Detail of Power behaviour with respect to traffic load/packet size.

7.1.4 Type of connections and relations with the other parts of the chain

Two 1850 TSS320 devices bridge the data path coming from CNIT1 and CNIT2 Routers. Resource availability is modulated according to day/night traffic profile, being able to switch on/off one of the two 10G data paths. This traffic requirement is propagated to the emulated network accordingly, where control plane allocates the paths according to an energy-based metric.

7.1.5 Metrics and parameter collected during the demonstration

During the demonstration, it was possible to collect data both from real nodes and from emulated networks, thanks to the node power model based on real measurements done on the actual device.

The details of the collected data are reported in the deliverable D6.5 “Benchmarking and performance evaluation results” [7].

7.2 Core Cloud

The core cloud is composed by two physical instances of the DROP router and by two emulation networks implementing green IP routing strategies. The remainder of this section is organized as follows. Section 7.2.1 describes the architecture and the functionalities of the two physical prototypes in the core cloud. Sections 7.2.2 and 7.2.3 describe the two emulation networks, the architecture of emulated nodes, and recall some details of the green routing algorithms. Section 7.2.4 deals with some integration aspects to interface the entities in the core cloud with the other devices and systems in the demonstrator. Section 7.2.5 gives a summary of the monitored performance indexes.

7.2.1 The CNIT1 and CNIT2 router prototypes

During the third year of ECONET, CNIT further extended the open-source software framework DROP (Distributed Router Open Platform) toward NFV/SDN architectures and power management capabilities. As already described in previous deliverable reports [3][4], DROP was originally designed as a middleware for the realization of extensible multi-chassis Linux software routers on top of Component-off-the-shelf hardware platforms [10], and for transparent integration of Linux network control-plane applications, like Quagga and Xorp. DROP aims at aggregating and autonomously coordinating all these building blocks into a single logical IP node, hiding the complexity of its modular architecture to control-plane applications and system administrators. In the new version DROP has been extended:

(i) to natively interface and integrate with OpenFlow switches within its modular architecture;
(ii) to provide an enhanced and even more transparent integration environment to native Linux control-plane applications and administration command-line tools (e.g., "ip");
(iii) to implement advanced power management strategies by means of the GAL, by integrating it at various levels of its architecture;
(iv) to easily integrate novel open-source frameworks for data-plane processing in the Linux user-space, like, among others, the Intel Data-Plane Development Kit (DPDK) [13] and Netmap [14], which are often applied to realize custom packet forwarding chain operations at very high speeds on general-purpose processors.

In few words, the DROP architecture has been specifically designed to act as "glue" among a large number of the most promising and well-known open-source software projects, providing novel data- or control-plane capabilities. To this purpose and as shown in Figure 25, DROP implements a number of "standard" interfaces native to the Linux operating system (e.g., Netlink, which is the standard Linux interface to notify configuration changes among control processes and the Linux kernel), to SDN devices (Openflow), and to the network device power management (i.e., GAL). These interfaces are used in the DROP southbound direction to transparently interact with multiple data-plane elements, as well as to provide a simplified device view at the control-plane processes in the northbound.

Between north- and southbound interfaces, DROP implements a number of mechanisms to manage the internal topology, to (dis-) aggregate network configuration data, like the Forwarding Information Database (FIB) or the power management configurations of elements, and to support slow-path communications (i.e., delivery of signalling traffic to and from the control-plane).

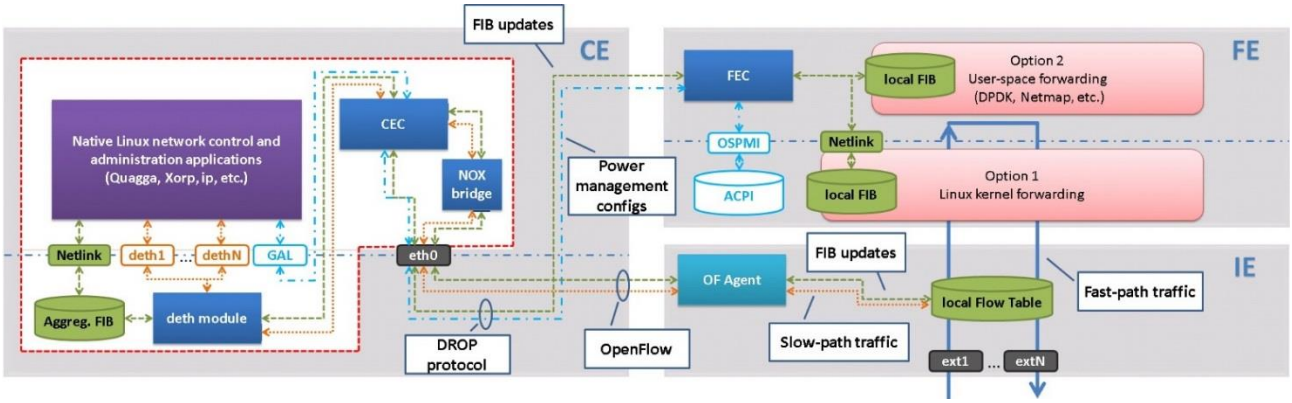


Figure 25: The DROP software structure including the main building blocks, interfaces, and slow- and fast-paths.

Figure 26 shows the physical setup of both routers, and in some sense the entire physical interconnection of the core cloud. In order to provide the needed number of ports with the requested speeds, two Interconnection Elements (IEs) have been included in the two routers. The first interconnection element is a SX60036 switch from MLX, and it is identical to the prototype used in the datacenter cloud. This switch is shared between the two router instances: ports 1-18 are owned by the CNIT1 router and ports 19-36 by the CNIT2 router. The second IE is a Pronto 3290 switch already used during the second year demonstration [1]. In this case, each router has its own IE.

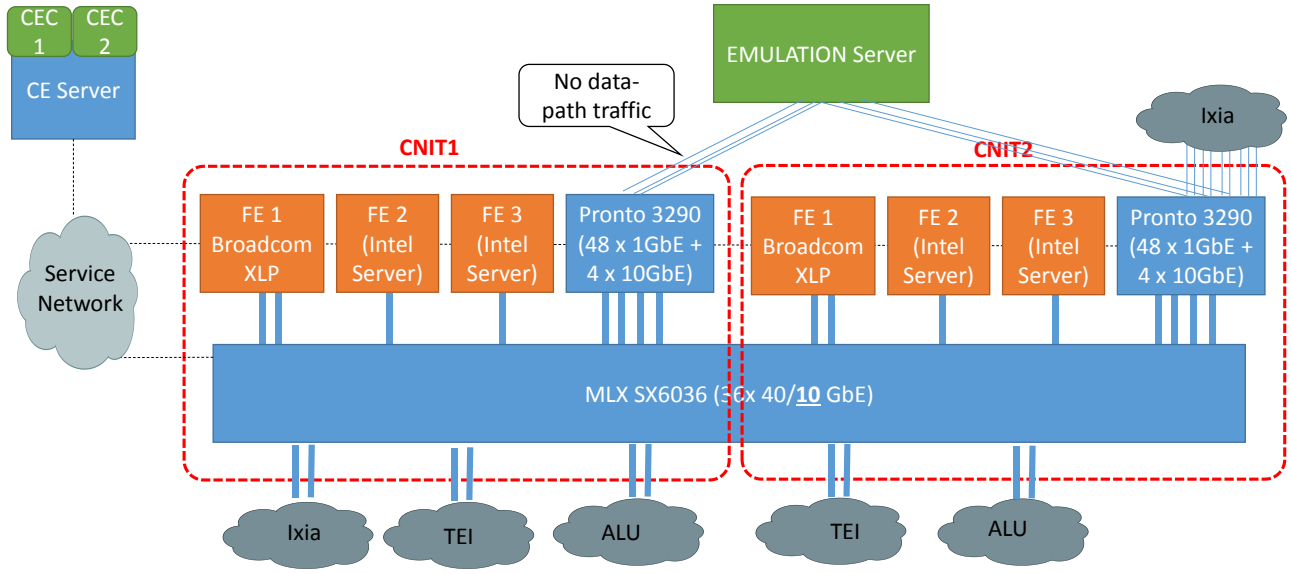


Figure 26: Physical setup of the CNIT1 and CNIT2 routers, physical interconnections and building blocks.

The two routers are orchestrated by two Control Element Controller applications, which have been installed on the same physical server connected to the out-of-band service network.

Each router instance is also composed by three Forwarding Elements (FEs) with the same hardware architecture deployed during the second year demonstration [1]. Two of them (FE1 and FE2) are realized with commercial off-the-shelf servers with processors of the family Intel Xeon 56XX. In more detail, FE1 has two 6-core processors working at 2.66 GHz, and the FE2 has two 4-core processors working at 2.40 GHz. The FE3 is realized with an evaluation board of the Broadcom

XLP832 System-on-Chip family, which is based on an 8-core MIPS processor with 4-way multi-threading support and with a maximum operating frequency of 1.5 GHz. Further details can be found in Sections 3.1.2 and 3.1.3 of the D6.2 report, and in Sections 5.1.1 and 5.1.2 of the D3.3 report.

Despite the hardware platforms are almost the same with respect to the second year demonstration, a huge effort has been devoted to the Software/Firmware level, in order to increase the DROP forwarding performance (for switching all the traffic volumes with the flexibility level required to support the final demonstrator definition – see Section 4), and in refining the power management support (i.e., LCPs and GAL).

One of the major evolutions of the new DROP version is related to the FEs. The packet forwarding is no more performed inside the Linux kernel, but performed at the user-space by using the Netmap framework in FE1 and FE2, and the Broadcom HyperApp libraries in FE3. It is worth mentioning that, with a joint effort between CNIT and MLX, the Linux driver of the MLX ConnectX3 adapter used in server-based FEs has been extended to support the Netmap framework. Both Netmap and HyperApp allow the access to the NIC hardware resources directly from a user-space application.

Therefore, two highly-optimized applications have been developed (the first one for the Netmap based FEs, and the second one for the XLP-based hardware). Despite the different I/O libraries and some low-level optimization details, the two applications share a same multithreaded architecture: a number of application threads, equal to the number of logical cores available on the FE, is allocated at the application bootstrap. Each application thread is bound to a couple of Reception/Transmission buffers of the NIC (or of the network accelerator in case of the XLP832 architecture). A generic thread works by looking if new packets arrived in the reception buffer; if so, it performs all the needed IP forwarding and lookup operations for each packet, and, finally, it places it into the transmission buffer. In both the used hardware architectures, the user-space forwarding allowed a forwarding performance improvement of a factor between 2 and 3.

The “affinities” (i.e., on which processor core a thread runs) of application threads are controlled by the second-level LCP, introduced later in this section. Thus, when this LCP decides to use a different number of cores, the cores are waken-up or put in standby mode by the LCP itself by means the GAL “Convergence Layer,” and the affinities of some of the threads are changed, in order to use also the waken up cores, or to free the cores going into standby modes.

Apart from these low-level integration details, the GAL tree architecture and the LCP algorithms have been preserved from the second year demonstration, so they still correspond to the situation depicted in the D4.3 and in the D6.2 reports, from whom Figure 27 is taken.

Therefore, a two-level hierarchy of LCPs is used to control the power configuration of DROP. The LCPs at the level of the forwarding elements orchestrate the AR and LPI capabilities to obtain two autonomic P-PsSs and one S-PsS (available only on commodity-based platforms). In more detail, as sketched before, this kind of LCP estimates the number of cores to be activated, and balances the threads of the user-space forwarding application among them. The unused cores are put into the ACPI C_6 state.

On its turn, depending on the incoming traffic volume, the LCP at the device level autonomously manages the EASs exposed by lower layer entities and the traffic share sent to each forwarding element, in order to obtain two autonomic P-PsSs, which are then decomposed into a couple of available EASs for each logical link configured on the device. To perform these operations, this LCP works by estimating how many FEs need to be active to serve the entire incoming traffic load, by waking up or putting them into standby modes, and consequently balancing the incoming load among active FEs.

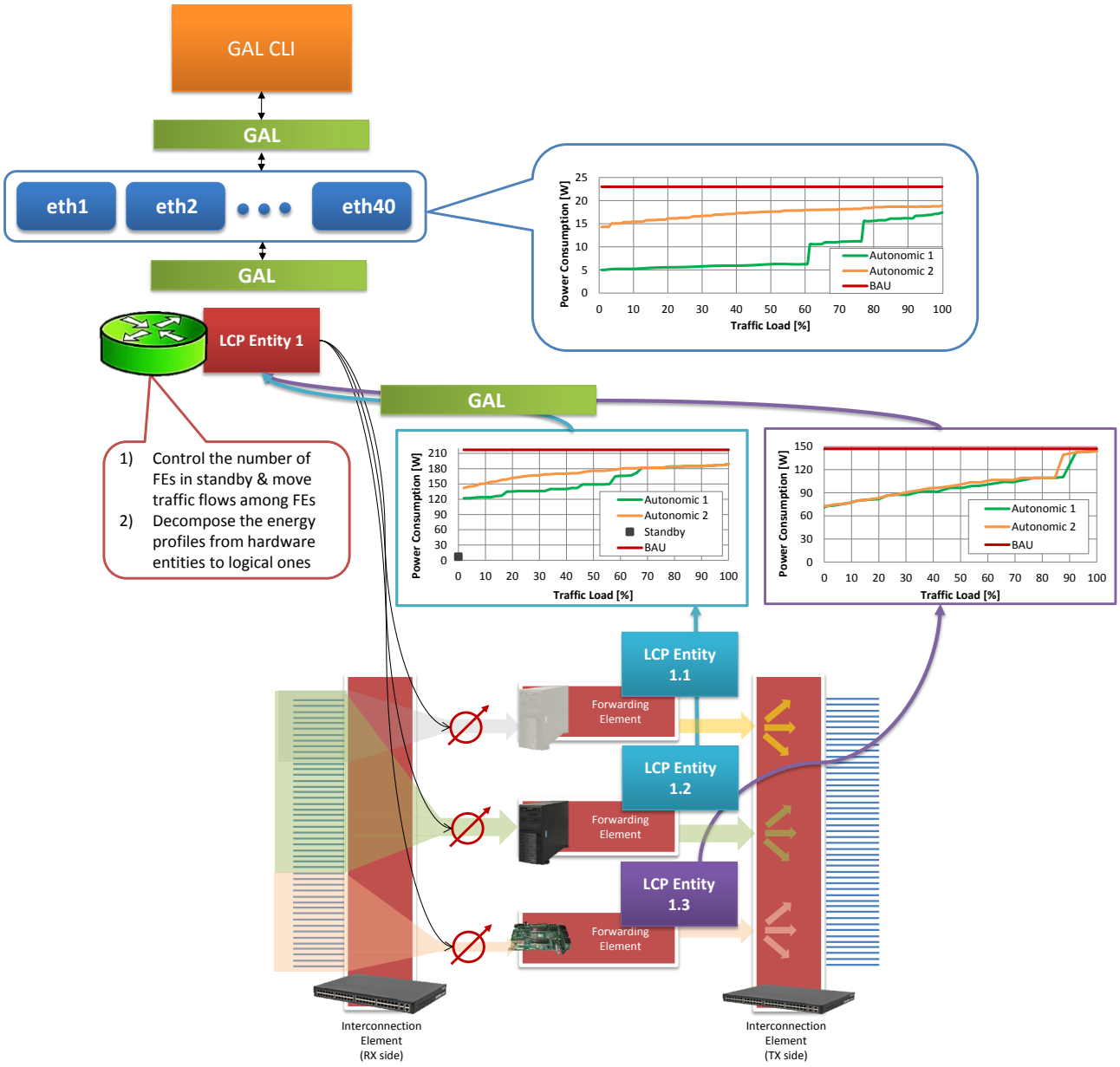


Figure 27: Overview of the role of LCPs in the DROP prototype and of the management of EASs at different hierarchical layers of the GAL.

7.2.2 The Emulation Network 1

The emulation network 1 includes the ESIR routing mechanism introduced in Section 4.3.5 of the D5.3 report [5].

The emulation environment was realized by means of the Netkit framework virtualization system, which allows to have multiple User-Mode Linux virtual machines interconnected among themselves with a very low computational overhead.

To realize the topology in Figure 4, 36 virtual machines have been set up, each of them including a modified version of the Quagga routing suite, which implements the aforementioned routing mechanism. The virtual machines, the interconnection among them, and the configurations of network interfaces and of the green OSPF protocol are generated through a set of automated scripts. The link characteristics and their capacity values are defined in Table 6.

In order to avoid scalability issues of the emulator, it has been decided not to generate data-path traffic among the nodes. This decision depends on the fact that generating data-path traffic on the emulated network causes a significant computational load on each User Mode Linux (UML) kernel, and consequently it may cause the saturation of the CPUs of the underlying hosting server.

Table 6. Definition of the links in the emulation network 1.

Link ID	*Source Node*	*Dest. Node*	*Capacity (Gbps)*	*Link ID*	*Source Node*	*Dest. Node*	*Capacity (Gbps)*	*Link ID*	*Source Node*	*Dest. Node*	*Capacity (Gbps)*	*Link ID*	*Source Node*	*Dest. Node*	*Capacity (Gbps)*
1	1	33	4.4	37	18	29	1.2	73	27	16	2.8	109	33	35	0.8
2	1	35	0.4	38	18	30	2.8	74	27	34	0.8	110	33	36	5.2
3	2	33	3.2	39	19	29	4	75	27	35	5.2	111	33	38	12.8
4	2	35	0.4	40	19	30	12	76	28	14	4.4	112	34	3	0.4
5	3	33	8	41	20	34	73.6	77	28	15	3.2	113	34	20	73.6
6	3	34	0.4	42	20	35	1.6	78	28	16	8	114	34	22	0.4
7	3	37	0.4	43	21	33	53.6	79	28	31	15.2	115	34	23	8.8
8	4	33	11.6	44	21	35	1.2	80	29	17	1.2	116	34	25	18.4
9	4	35	0.4	45	22	33	15.2	81	29	18	1.2	117	34	27	0.8
10	4	37	0.4	46	22	34	0.4	82	29	19	4	118	34	31	58.8
11	5	23	3.2	47	23	5	3.2	83	29	33	0.8	119	34	33	1.6
12	5	24	9.2	48	23	6	2	84	29	35	5.2	120	34	35	2.4
13	6	23	2	49	23	7	4.4	85	30	17	2	121	34	36	10
14	6	24	4.8	50	23	8	0.8	86	30	18	2.8	122	34	38	25.6
15	7	23	4.4	51	23	33	0.8	87	30	19	12	123	35	1	0.4
16	7	24	14.4	52	23	34	8.8	88	30	31	16.4	124	35	2	0.4
17	8	23	0.8	53	24	5	9.2	89	31	24	30	125	35	4	0.4
18	8	24	2	54	24	6	4.8	90	31	26	61.2	126	35	20	1.6
19	9	25	1.2	55	24	7	14.4	91	31	28	15.2	127	35	21	1.2
20	9	26	2.4	56	24	8	2	92	31	30	16.4	128	35	27	5.2
21	10	25	2.4	57	24	31	30	93	31	33	73.6	129	35	29	5.2
22	10	26	5.2	58	25	9	1.2	94	31	34	58.8	130	35	31	0.4
23	11	25	6.4	59	25	10	2.4	95	31	35	0.4	131	35	33	0.8
24	11	26	20.8	60	25	11	6.4	96	31	37	0.4	132	35	34	2.4
25	12	25	2	61	25	12	2	97	32	36	18	133	35	36	2.8
26	12	26	5.6	62	25	13	8.8	98	33	1	4.4	134	35	38	6
27	13	25	8.8	63	25	33	1.6	99	33	2	3.2	135	36	32	18
28	13	26	28	64	25	34	18.4	100	33	3	8	136	36	33	5.2
29	14	27	2	65	26	9	2.4	101	33	4	11.6	137	36	34	10
30	14	28	4.4	66	26	10	5.2	102	33	21	53.6	138	36	35	2.8
31	15	27	1.6	67	26	11	20.8	103	33	22	15.2	139	37	3	0.4
32	15	28	3.2	68	26	12	5.6	104	33	23	0.8	140	37	4	0.4
33	16	27	2.8	69	26	13	28	105	33	25	1.6	141	37	31	0.4
34	16	28	8	70	26	31	61.2	106	33	29	0.8	142	38	33	12.8
35	17	29	1.2	71	27	14	2	107	33	31	73.6	143	38	34	25.6
36	17	30	2	72	27	15	1.6	108	33	34	1.6	144	38	35	6

7.2.3 The Emulation Network 2

The emulation network 2 includes the GRiDA routing mechanism introduced in Section 4.3.1 of the D5.1 report [6]. It runs on the topology depicted in Figure 5, with the following link characteristics:

- L-1: bidirectional link between nodes 1 and 2, link capacity equal to 30 Mbps, power consumption of the interfaces equal to 30 W;
- L-2: bidirectional link between nodes 2 and 4, link capacity equal to 30 Mbps, power consumption of the interfaces equal to 30 W;
- L-3: bidirectional link between nodes 1 and 3, link capacity equal to 5 Mbps, power consumption of the interfaces equal to 5 W;
- L-4: bidirectional link between nodes 2 and 3, link capacity equal to 10 Mbps, power consumption of the interfaces equal to 10 W;
- L-5: bidirectional link between nodes 4 and 6, link capacity equal to 30 Mbps, power consumption of the interfaces equal to 30 W;
- L-6: bidirectional link between nodes 3 and 5, link capacity equal to 10 Mbps, power consumption of the interfaces equal to 10 W;
- L-7: bidirectional link between nodes 5 and 6, link capacity equal to 30 Mbps, power consumption of the interfaces equal to 30 W.

Moreover, a base power consumption of 100 W has been associated to each virtual node.

The emulation environment was realized by means of the VirtualBox virtualization framework, which allows to have complete Ubuntu Linux virtual machines interconnected among themselves.

As in the previous emulation network, 4 virtual machines have been set up to realize the topology in Figure 5. Each virtual machine includes a modified version of the Quagga routing suite, which implements the aforementioned routing mechanism. Additional details on the software implementation of GRiDA in the Quagga routing suite can be found in Section 4.3 of the D5.3 report [5]. Also in this case, the virtual machines, the interconnection among them, and the configurations of network interfaces and of the green OSPF protocol are generated through a set of automated scripts.

Since the GRIDA mechanism implementation requires the presence of real data-path traffic, it has been decided to generate it by other two virtual machines acting as traffic source and sink, respectively. To avoid the scalability issues of the emulator, the topology has been built with few nodes and the traffic generated is very low (e.g., 30 Mbps at maximum).

7.2.4 Integration and External Interfaces

As previously sketched, and as shown in Figure 28, the two physical prototypes have been mapped into the emulation networks. The integration between the green routing algorithms and the DROP prototypes has been performed with the clear objective to demonstrate the feasibility of the ECONET architecture, and of the use of the GAL in IP networks. This mapping has been done by:

- installing the modified versions of the Quagga routing suite[1] into the CECs of the CNIT1 and CNIT2 routers,
- configuring them in order to own the correct list of IP interfaces;

[1] It is worth mentioning that CNIT1 and CNIT2 run exactly the same software application version running on the virtual machines in the emulation networks.

- suitably mapping these interfaces on a set of ports of the IE2 of each router (i.e., the Pronto switch – see Figure 26);
- connecting accordingly these ports to the emulation server, where virtual routers run;
- configuring the emulation environment in order to map the link between a virtual node to the Quagga instance installed on a physical prototype on a specific physical interconnection link between the CECs and the emulation server.

It is worth recalling that DROP1 and DROP2 have been mapped to nodes 25 and 26 of the emulation network 1 (see Figure 4), and to nodes 1 and 6 of the emulation network 2 (see Figure 5), respectively. So, CNIT1 has 7 "virtual" links interconnecting it to virtual nodes of the emulation network 1, and 2 links toward virtual nodes of the emulation network 2. CNIT2 has 6 links toward the emulation network 1, and 2 toward the emulation network 2. Consequently, the emulation server has been equipped with 17 interfaces, connected to 9 ports of the IE2 of CNIT1 and 8 ports of the IE2 of CNIT2.

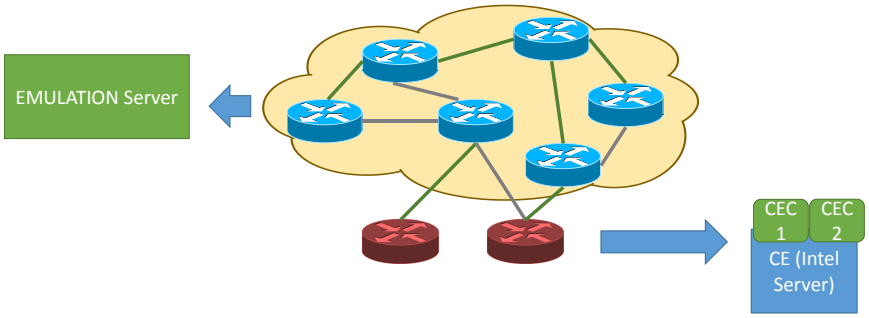


Figure 28: Mapping of the CNIT1 and CNIT2 prototypes in the emulation networks for evaluating the green routing extensions.

All the physical and virtual nodes in the core cloud have been fully integrated with the green monitoring and management system running in the 7th demonstration cloud (see Section 5). To this purpose, both the server hosting the CECs of the DROP routers and the one hosting the emulation networks have been equipped with an additional interface connected to the out-of-band signaling network, where also the monitoring and management system is connected.

Two main kinds of performance indexes are exposed by physical and virtual nodes; namely, the GAL states and sensors through the GAL REST protocol, and the traffic (transmission and reception) counters for each link through the SNMP IF MIB (Simple Network Management Protocol – the InterFaces group – Management Information Base) standard interface [15].

Regarding the CNIT1 and CNIT2 routers, as shown in Figure 29, the DROP software framework has been extended to integrate a complete implementation of the GAL REST protocol. In detail, as shown in Figure 30, the GAL REST interface exports all the entities from layer 0 (the one with logical resources) up to the layer 2 (the layer with FEs and IEs). Despite the information on the entities and related energy-aware states, the GAL REST implementation exports a sensor with the energy consumption of the entire router or of single (physical) entities. Through the GAL REST, it is also possible to set/change the committed energy-aware states of the entities at different layers.

It is worth mentioning that, since the MLX switch has a GAL implementation on board, it has been interfaced with the DROP CE (thanks to a joint effort between CNIT and MLX), which is now capable of exposing its states and the values collected by its internal sensors. Unfortunately, given the final

dimensioning of the demonstration network, it was not possible to trigger the power management mechanisms in the MLX switch (for triggering these mechanisms a link capacity of 40 GbE is needed).

The Net-SNMP software framework (an open source implementation of SNMP and of the IF MIB) [16] has been installed on the CECs, and the DROP CE application extended for aligning the traffic counters of the network interfaces exposed by the operating system (named "Virtual Interfaces" in Figure 29), with the ones of the ports of IEs, where "public" links (i.e., the links toward other devices) reside. This way, the Net-SNMP application can natively collect the values of those counters and export them to the monitoring and management system.

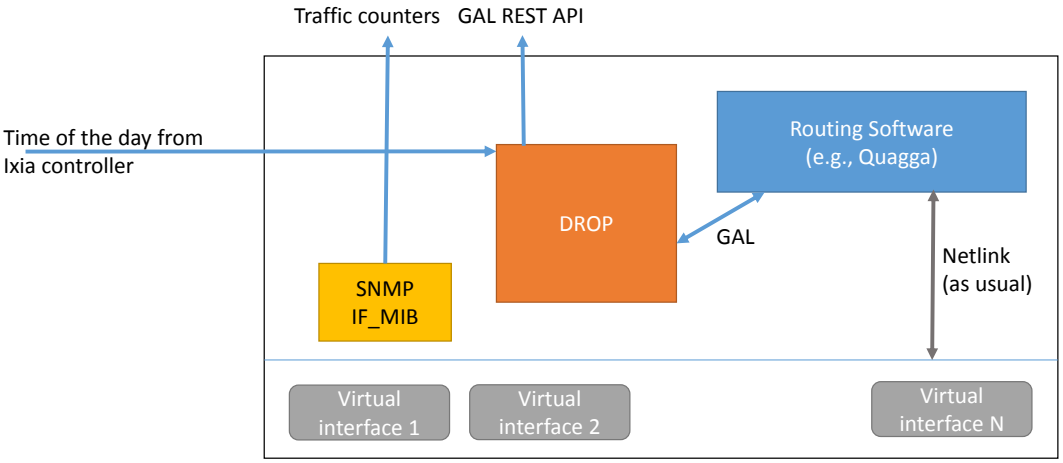


Figure 29: Integration of the external interfaces in the CNIT1 and CNIT2 nodes.

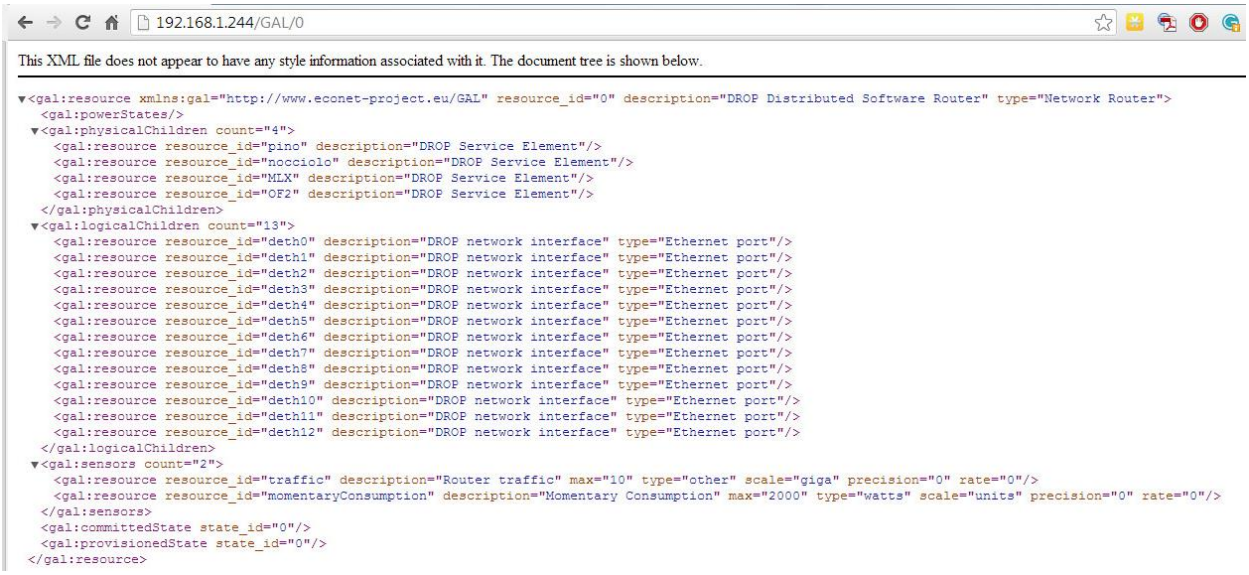


Figure 30: Screenshot of a web browser accessing the GAL REST interface (at the entity 0) of a physical DROP prototype.

In order to make virtual nodes exposing the same data set to the monitoring and management system, joint integration and development activities among the two CNIT units (the one in Genoa and the one in Turin) and the collaborating institution of the University of Rome "La Sapienza" have been performed. These activities led to the software architecture depicted in Figure 31.

In more detail, a patched and simplified version of the DROP CE application, named "virtual DROP", has been developed in order to provide the GAL interface to the Quagga routing suite, and translate it in the GAL REST protocol interface. This version of the GAL (and of the GAL REST) only exposes the state of logical resources (i.e., the links of the virtual node) and the root entity. A virtual sensor reporting an estimate of the power consumption of the node has been included in the root entity. Differently from the complete DROP CE, the virtual DROP does not include any LCP or logic to manage subsystems: it can be considered only a simple cross-protocol/API interface.

Moreover, as discussed in Section 7.2.2, some additional integration activities were needed in the emulation network where data-path traffic has not been generated. In such a case, a further software application has been developed for emulating the increase of network interface traffic counters as the data-path traffic really flows through the network interface.

To this purpose, and without entering into a too detailed description, this application has been triggered by the "time of the day" message sent by the Ixia traffic generator, in order to update the increased rate according to pre-defined traffic matrixes. The results are written on a file, which is read by a version of the Net-SNMP application, which was appropriately patched to collect counter values from this new file, rather than from the local operating system.

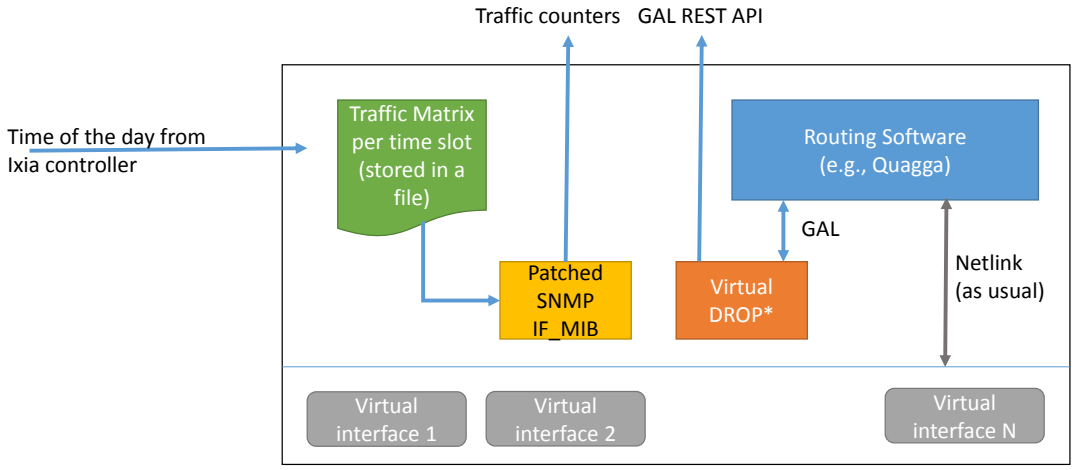


Figure 31: Integration of the external interfaces in the virtual nodes.

Finally, as already sketched in Section 5, a further integration activity has been done between the router and the transport cloud. In order to make CNIT1 and CNIT2 routers able to change the light-path reservation from the day configuration to the night one, and vice versa, the DROP CE has been extended to send Transaction Language 1 (TL1) messages to the transport network controller to inform it about the routing change.

7.2.5 Metrics and parameter collected during the demonstration

The following performance indexes and information are exposed from each physical or virtual node to the green monitoring and management framework:

- The GAL tree and all the contained entities;
- Committed and available EASes (the virtual nodes only have 2 states per link corresponding to the fully active and standby states);
- Energy consumption of all the GAL entities residing at layers equal to or greater than 1;
- Received, transmitted and lost Bytes for each network link;
- Received, transmitted and lost packets for each network link.

In addition to the previous list, the physical prototypes expose the following indexes to the DROP GUI (Graphical User Interface):

- Volume of traffic sent to a FE port;
- Number of cores left active in each server-based FE;
- ACPI P- and C-states committed/available on each core of the processors in the server-based FEs;
- Frequency and supply voltage of each MIPS core in the XLP 832 SoC.

7.3 Metro transport

TEI's Energy Aware Networking is a set of energy saving features empowered by the GAL**.** The Energy Aware Traffic Engineering developed in the course of this study routes traffic according to the load and taking into account the "power costs" of the available resources, to minimize power consumption while guaranteeing QoS.

TEI developed the Energy Aware Networking on its POTP Metro Node platform SPO1460.

Figure 32. the POTP Metro Node platform SPO1460.

7.3.1 The Network Management System

In the Energy Aware Networking, TEI developed a NMS (Network Management System) aware of the topology, of the traffic load, of the power management capabilities and of the power state of each individual element in the Network.

Governed by GAL rules and by local controlling intelligence (LCP, Local Control Policy) and/or by centralized controlling intelligence (NCP, Network Control Policy) at the management layer, the system can route traffic load towards certain resources and put the remaining resources to sleep, by applying strategies like Voltage Scaling, Frequency Scaling, Selective Power Off, etc.

The main features of the prototype are:

- Energy Aware Link Aggregation Group (LAG).
- Energy Aware Routing.

7.3.1.1 Energy Aware LAG

A standard LAG spreads active traffic on all available links ignoring power consumption. As a result, even in the presence of low traffic load, all links are up and fully powered.

The Energy Aware LAG is a smart algorithm, patented by TEI, which allows dynamically concentrating the incoming traffic on a lower number of ports, and switching off unnecessary links when the traffic is sufficiently low.

The EA LAG is an example of Local Control Policy.

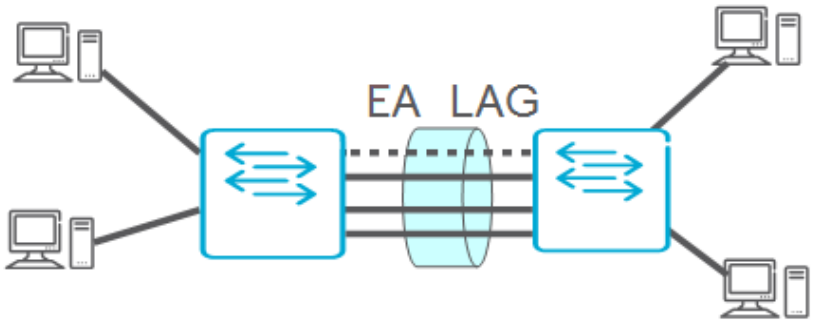


Figure 33: The Energy-aware LAG.

7.3.1.2 Energy Aware Routing

In the presence of low loads, traffic is directed over pre-selected resources, while the remaining ones are forced to enter deeper Sleeping modes:

- Fast Sleep → no traffic, good power saving, some computation capability, wake up time in milliseconds
- Deep Sleep → no traffic, very good power saving, extremely poor computation capability, wake up time in seconds
- Power Off → no traffic, almost null power consumption, no computation capability, wake up time in 5 to 6 minutes

The EA Routing is an example of Network Control Policy.

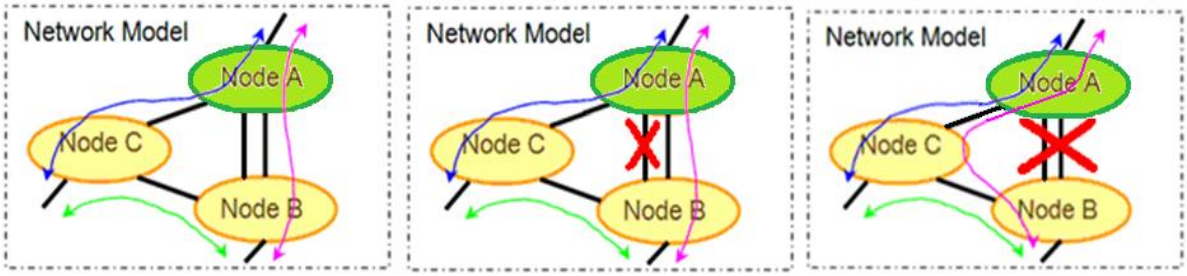


Figure 34: The Energy-Aware Routing in the Metro ring.

7.3.2 Device characteristics with respect to the transmission and energy aspects

The Energy Aware Networking prototype shows a clear dependency of consumed energy as a function of the transported traffic load.

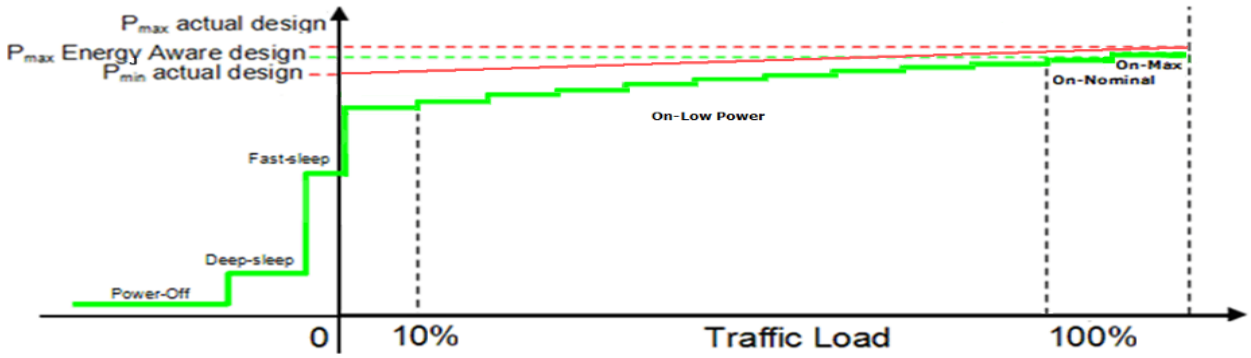


Figure 35: Power profile of the energy-aware Metro prototype.

The number of available resources is reduced according to the current requirements, and resources can be dynamically resumed in case of traffic increases.

The power consumption has been measured at full traffic load and at ~10% decremented traffic load steps, i.e. 9G, 8G, 7G, … until the no-traffic-load condition.

7.3.3 Metrics and parameters collected during the demonstration

The power consumed at various traffic load levels has been collected. Such values have been compared with the ones obtained on an actual Network dimensioned as follows:

- peak rate ~7G.
- over-provisioning + redundancy = 20G (from 25% to 50% with respect to the estimated peak load).
- dimensioning mean utilization ~3G, which represents ~15% of the Network's capabilities utilization.

The obtained results give the clear indication that ECONET and Energy Aware Networking hold the promise to enable extremely aggressive power saving, well above 50%.

We can also correlate collected numbers with some meaningful indexes, namely OPEX saving and CO_2 emissions, and estimate the related saving obtained by the massive application of Energy Aware Networking:

- Assuming the whole Transport portion of a nation-wide European Operator based on SPO1400, the Energy Aware Networking gives around 15M€ saving per year.
- Assuming the whole Network of a nation-wide Operator based on Energy Aware Ericsson equipment gives around 200M€ saving per year.
- Assuming the whole Transport portion of a nation-wide European Operator based on the Energy Aware SPO1400 gives a reduction in the carbon footprint equivalent to around 300K cars less per year.
- Assuming the whole Network of a nation-wide Operator based on Energy Aware Ericsson equipment gives a reduction in the carbon footprint equivalent to around 4M cars less per year.

7.4 Data centre

The growth in Cloud and Web 2.0 storage and computing requirements in recent years has led to an increase in demand for larger, stronger, and more cost-efficient datacenter.

As datacenter are growing in size, the interconnecting fabric is becoming larger and more complex, resulting in more power consuming network equipment, both in the core and on the edges.

This higher power consumption increases both operational costs and carbon footprint, increasing the significance of traditionally overlooked power aspects. Furthermore, rarely all nodes in the data centre are active, requiring full network connectivity to be maintained 24/7. It should be noted that even active nodes do not carry full BW operation during all day long. On the contrary, there is a large portion of time in which links carry less than the full capable BW. Presently, many data centre devices (switches, NICs, etc.) use the same power, no matter what is the BW used on each port. One of the aims of the low power features incorporated in the data centre devices is to lower the power of the devices once lower BW traffic is used on each port.

An adaptive fabric, capable of shutting down currently unused network elements and self-optimizing its topology, will increase energy efficiency, decrease CO_2 emissions, and reduce the data centre cost of operation.

The “green” fabric concept presents the Mellanox power-efficient features under development as part of the ECONET project.

Mellanox low power features have the capability to reduce power consumption by up to 43%. When summed over a real-world data centre scenario, a total reduction of 13% of all network components’ power consumption is demonstrated. This reduction can amount to millions of dollars in savings over several years.

7.4.1 Function of the device in the network

Mellanox has implemented the emulation of a data centre cloud in the ECONET network. A typical data centre includes servers connected between themselves and the network via Network Interface Cards (NICs) and Switch devices. Mellanox generates both of these types of devices. For the data centre cloud emulation, Mellanox used an HP D640 server connected via ConnectX™-3 NIC card to a VPI switch SX1036. The switch emulated the data centre connectivity network by performing Snake

connectivity topology. In the Snake topology, the traffic in injected on one side of the snake, and it traverses through all the ports in the switch which are connected one to the other: port 3 connected by cable to port 4, and then from port 4 there is an internal connection to port 5 via explicit VLAN number, while port 5 is connected by cable to port 6, etc. This topology will enforce the traffic to propagate through all the switch ports; hence, it emulates the transition between different switches in the data centre fabric. The last port is connected via cable to the server. The server was running a UDP reflector program to inject the packets back on the snake topology.

By throttling the amount of traffic that passes through the snake topology, the low power features of the data centre emulation topology can be expressed, and the consumed power of the switch can be measured. Refer to Figure 2 for reference on the connectivity topology.

7.4.2 Type of technology used

The MLX devices incorporate the following low power features:

- Smart Clock gating: every cell in the ASIC design incorporates clock gating controlled by smart enable signal that tracks the activity and BW delivered by the cell.
- Voltage scaling: enables to control the main V_{DD} voltage of the chip and decrease the power consumption by doing so. Whenever the chip's effort is not at its maximum and lower V_{DD} voltage is sufficient to support its operation (e.g., to lower core PLLs usage), the Voltage Scaling feature will recognize it by using its smart monitor and will lower the input voltage, as described in D3.3, Section 5.1.1.1.2.2.
- Frequency scaling: Mellanox ASIC devices are capable of operating at different core frequencies by providing different protocol capabilities, as well as different performance points (e.g., different latency and packet processing rate). By configuring the device to work at lower frequency, it is possible to reduce the system's power consumption while reducing the devices performance, as described in D3.3, Section 5.1.1.1.2.2.
- HW shutdown: In order to enhance the power saving, MLX also developed HW shutdown support for cases of unused internal modules within director switches, as described in D3.3, Section 5.1.1.1.2.2. This is relevant for cases where entire ports are not used (module not present/ port disabled / line card disabled/ etc.); hence, it is not included in the ECONET demonstrator, since in the snake topology all ports are active. This was done to show maximum power reduction operation based on tracking the BW traffic on all the ports.
- Power-Aware PHY optimization: enables automatic shutdown of the non-active logic, including state machines, PLL units, non-active SerDes modules based on the PHY protocol in use, as stated in D3.3, Section 5.1.1.1.2.3.
- Heat dissipation and Fan control, including smart cooling control algorithm: A power aware fan algorithm to support effective heat dissipation, while minimizing the power consumption of the entire system, as described in D3.3, Section 5.1.1.2.3.
- Width Reduction Power Saving (WRPS): Modern link technologies are using multiple lanes to scale link bandwidth. An example would be the 40GBASE-KX4 40GigE Ethernet protocol using 4 lanes of ~10GHz transmitters and receivers to reach the 40GigE capacity. It is known that the transmitter / receiver consumes a considerable amount of power even when data is not transmitted over the link, while transmitting IDLE symbols used to synchronize the transmitter and receiver bit alignment and skew. WRPS technology enables reducing the number of active lanes on a link (and turning off the unused transmitters and

receivers) dynamically without interruption of the packet flow over the link. For example, on a 40GigE link, it is possible to turn off 3 of the lanes and still support a capacity of 10GigE, while reducing the number of active transmitter / receiver pairs by 75%. The WRPS feature tracks the BW traversing each port and operates based on hysteresis band: when traffic goes over 10Gbs, all four lanes are enabled, while when traffic goes back below 8Gbs three lanes will be turned off and the link will operate in single lane mode. This feature is described in detail in D3.3, Section 5.2.1.2.

The Mellanox switch supports two power modes, which can be controlled by the GAL mechanism:

- Power_mode_1: WRPS feature disabled
- Power_mode_2: WRPS feature enabled

During the demonstration, the system was performing in those two modes by control of the system power controller, and power metrics were captured.

7.4.3 Characteristics of the device with respect to both transmission and energy aspects

The ECONET demonstrator uses MLNX switch type SX1036. This switch has 36 ports of VPI connectivity, i.e., it can operate both at Ethernet protocols or InfiniBand protocols. For the Ethernet protocol, it can run up to 40 Gbps Ethernet, and for the InfiniBand protocol, it can run up to 56Gbps. In the ECONET demonstrator, all links were used in Ethernet mode, i.e., with max 40Gbs max traffic BW per port.

Since the WRPS feature operates on the granularity 1X operation or 4X operations, then for 40Gbps links the transition point between 1X operation and 4X operation is 10Gbps BW.

The MLNX switch tracks the traffic bandwidth on each port and optimizes the device characteristics for power consumption. Figure 37 describes the change in total system power based on port traffic usage. The SX1036 switch was used with increased BW load on all the ports, from 0Gbps up to the max 40Gbps, while the power is low for rates up to 10Gbps, and increases for higher rates than 10Gbps, due to the transition from 1X operation to 4X operation:

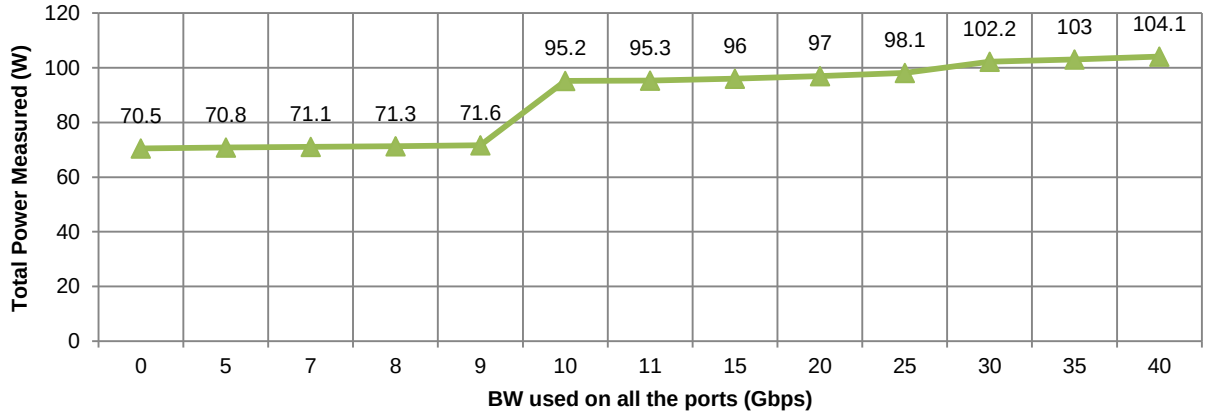


Figure 36: Energy consumption of the MLX device working in System "Power_mode_2" with WRPS enabled.

7.4.4 What type of connections and relations with the other parts of the chain

The emulated data centre switch was connected with two links to the Metro cloud, L16, L17. Each of those links had a max traffic BW of 5Gbps, hence in total max of 10Gbps. This amount of traffic was not enough to demonstrate the operation of the WRPS feature, since the WRPS feature needs to throttle based on hysteresis band traffic of 8Gbps/10Gbps, as described above. For this purpose, a third link was injected into the snake topology as a third traffic load, named F7. This link was injected from a Spirent traffic generator, to have the total snake traffic to throttle in a range exceeding the WRPS hysteresis band.

On the other side of the snake, the traffic was directed to a HP server, receiving a link of the aggregated traffic. This link has the capability to handle up to 40GbE traffic.

7.4.5 Metrics and parameters collected during the demonstration

During the demonstration, the switch was operated in two power modes. In Power_mode_1 the WRPS feature was not activated, while all the other power features (like automatic clock gating, voltage/frequency scaling, smart thermal control) where activated. In Power_mode_2 the WRPS feature was also activated on top of the rest of the features. The following metrics were observed during the traffic change in each mode:

- Total switch power.
- Switch power state/mode.
- Total traffic used by each link and total traffic propagated through the snake.
- ASIC thermal point under each traffic load.
- Fan speed under each traffic load.
- Number of lanes used in each port, tracked by RX_POWER_UP and TX_POWER_UP fields. This metric relates to the width used on each port, either 4X operation or 1X operation. It will change based on the Width Reduction Power feature described in D3.5 [17].

7.5 Access

Ethernity covered the access aggregation that requires certain power scaling functions, to accommodate changes in the traffic transmitted towards the user and received from the user, to enable monitoring of the traffic and change the device configuration in amount of ports, interfaces and functionality. In business CPE equipment the needed solution shall go more towards active sleeping, to enable shutdown functionality and disable inactive ports.

7.5.1 Function of the device in the network

Ethernity installed two different products in the demo network, one being 48-port VDSL DSLAM, and a business CPE with 8 GbE ports.

7.5.2 Characteristics of the device with respect to both transmission and energy aspects

The power scaling delivered by Ethernity for the demo was integrated into a 48-port VDSL DSLAM including three main power scaling functions, number of port scaling, functionality scaling, and buffer management scaling. For the power scaling solution, for the demo performed, Ethernity has implemented a new mechanism to enable dynamic power scaling functions, to consume low power during the low traffic scenario, which uses internal buffers together with the elimination of the use of the switching engine on the device functions. This way, all traffic coming from users will pass

transparently towards the network without implementing local switching and will use internal memory for buffering without the need to access the external memory. This design was implemented to support dynamic power scaling versus the design done during WP3, which targeted static scaling, by downloading different FPGA firmware to the FPGA, and with the target to provide a power-scaling solution capable of operating on ASIC and not just on Programmable devices (FPGA).

To support that dynamic power scaling ETY designed internal buffers that use the FPGA (or ASIC) internal memory resource to buffer traffic in low throughput scenarios during non-active periods. In that configuration the external DDR is in non-active mode and does not consume power, together with the switching engine block that is in idle mode. The design enabled us to provide a power scaling solution that can be used in both ASIC and FPGA. Over the DSLAM, we also enable the ability to shut down specific VDSL ports to decrease the power.

Ethernity also provided a business CPE solution with 8-GbE ports implementing some unique active sleeping technologies, such as disabling the transmit direction and leaving only the receive direction of a port to listen to incoming traffic, and upon that opening the transmit direction.

7.5.3 Characteristics of the device with respect to both transmission and energy aspects

On the 48-port DSLAMs installed for the demonstrator, we had the following power scaling features:

- Disabling the DDR components on the board and using internal ASIC memory, so that above 2Gbps of traffic passing through the internal buffers, the ASIC (FPGA) can detect the rate and switch to using the external DDR to support the required traffic aggregation up to 100ms buffering.
- Disabling the switching engine.
- Disabling a line (VDSL modem); this shuts down completely D/S and U/S traffic via this line.

Table 7: Power characteristics of the 48-port VDSL DSLAM ETY1.

Scenario	energy consumption [W]	Energy consumption with respect to the BAU case [%]
Board Business as usual	46.6	100
Board Power [W] with Internal Buffers:	45.6	97.9
Board Power [W] stop 16 lines:	41.8	89.7
Board Power [W] stop 32 lines:	38.4	82.5
Board Power [W] stop 48 lines:	35.0	75.3
DDR power	0.96	
Power per line (modem)	0.22	
FPGA power BAU	5.76	
FPGA power ECO mode	4.8	83.3

On the business CPE platform we had the following power features:

- Disabling the DDR components on the board and using internal ASIC memory, so that above 2Gbps of traffic passing through the internal buffers, the ASIC (FPGA) can detect the rate and switch to using the external DDR to support the required traffic aggregation up to 100ms buffering.
- Disabling the TX direction of a SerDES port (up-to 4GbE ports).
- Disabling the PHY of the 4 RGMII ports on our external Piggy card (up-to 4 GbE ports).

Table 8: Power characteristics of the CPE switch ETY2.

Scenario	energy consumption [W]	Energy consumption with respect to the BAU case [%]
Power [W] while FPGA load:	11.0	-
Power [W] without Internal Buffers (all 8 ports):	24.5	100
Power [W] with Phy_disable (4 ports)	21.1	86.3
Power [W] with Phy_disable (4 ports) + 2 ports tx_disable	19.2	74.4
Power [W] with Phy_disable + 2 ports tx_disable + internal buffers	18.2	64.2
Power [W] only FPGA + board components (with no Piggy card)	15.4	
Power [W] only FPGA + internal buffers	14.4	
Phy_disable (per port)	0.84	
TX_disable (per port)	0.96	
DDR power	0.96	
FPGA power BAU	4.32	
FPGA power ECO mode	3.36	77.8

7.5.4 What type of connections and relations with the other parts of the chain

1GbE of the 48 ports DSLAM was connected to the Ericsson switch and 3 VDSL ports out of the 48 VDSL ports available were connected to Lantiq VDSL home gateways. The Business CPE was connected to the Ericsson switch through 1 GbE port.

Moreover, both the ETY1 and ETY2 devices have been equipped with an implementation of the GAL REST protocol to report the energy-aware states to the central Green OAM framework. This implementation has been carried out in strict collaboration with CNIT researchers. In more detail, a further version of the virtual DROP application (see Section 7.2.4) has been developed in order to collect raw power management information from ETY devices, translate it into the embedded GAL implementation and expose it into the GAL REST protocol.

7.5.5 Metrics and parameter collected during the demonstration

The following parameters are collected during the demonstration:

- Sent and received packets/bits on each port;
- Status of each port (by means of the GAL REST protocol);
- Power consumption of the devices.

7.6 Customer network

The access network contributes the largest part of the energy consumption in the network. ICT equipment at customer sites accounts for the largest part of power consumption ascribable to this world, as it is deployed in millions of households and it is operated by non-technical experts, who behave mostly emotionally and inconsequently. Studies have shown an inefficient usage of such devices, which are left powered on even if unused; among other reasons, this is often done to maintain their presence on the network and let them easily reachable. Indeed, most of the tasks that such devices perform for this purpose consists of simple and repetitive actions that can be easily demanded to a specific function running always on low-power devices; according also to other authors, we refer to such function as *Network Connection Proxy* (NCP), and we thoroughly investigated it in this project [1] [4] [18].

The main assumption behind the NCP concept is the availability of a low-power device that is always on to provide this service. We think that, in a home network scenario, this role could be

suitably played by home gateways, as of their increasing networking and processing capability and their placement at the boundary between the user premises and the access network. The requirement for the home gateway to be always on leads to the necessity for the device itself to be highly power-efficient, and so to support embedded and autonomous energy consumption optimization policies. Therefore, the device itself must do the job of energy consumption minimization automatically without human intervention.

More and more the home gateway is becoming the central controller of the home. The device landscape in the homes is changing. PCs are replaced by tablets and the remaining ones are used for professional work only. The Set-top box is replaced by smart TV and web radio. None of these devices is always on. Smart grid services such as meter readout and home control must run 24 hours per day, 7 days a week over the whole year. The only device in the home, which can do the job, is the home gateway.

Taking into account that home devices are sometimes switched off or powered down – tablets are often out of battery – it makes sense that the home gateway assures the network presence for these devices when they are not in operation. The obvious example is Skype, where the proxy could display the "absent" state, which allows receiving messages, which is not possible in the "offline" state.

The combined support of NCP and autonomous energy consumption optimization policies within the home gateway can reduce both network traffic and power requirements in the access segment (indeed, the home gateway itself is part of the access network and is deployed in millions of customer sites). Moreover, it represents a basic step towards the integration of more synergies between the customer and access networks, according to the main philosophy of the project.

In this Section, we describe the contribution to the demonstrator relevant to the home gateway autonomous power saving policy (Section 7.6.1) and the NCP (Section 7.6.2).

7.6.1 Home gateway power management daemon

LQDE decided to go for a self-optimizing home gateway. It was realized by a power management daemon (PMD), a Linux program running in the background and optimizing continuously the energy consumption according to the currently required performance. The realization is described in detail in D3.3.

To demonstrate the practical usage of the approach LQDE provided three different home gateways, which may work with three Energy-Aware States that were labelled with the traffic light colours red, yellow and green (see Table 9).

Table 9. Energy-Aware States available in the LQDE HGs

Home Gateway	Power consumption average	Quality of Experience QoE
Red	Maximum	High
Yellow	Close to minimum	High; no noticeable difference to RED
Green	Minimum	Poor: slow browser, voice quality degraded

From the comparison above, it can be seen that the yellow home gateway has all advantages, low energy consumption with virtually no drawback in QoE. The low average power consumption of the home gateway is because a home gateway always has long periods of low usage, for example in the night or during the day when people are at work, or during vacation periods. A qualitative example is given Figure 37. In this example it is shown how the average power consumption of the yellow home gateway at the beginning, without any user traffic, is close to the minimum value of the green home gateway. During the activity period, the average power consumption increases and in the following idle time the average goes down again and approaches slowly the minimum value.

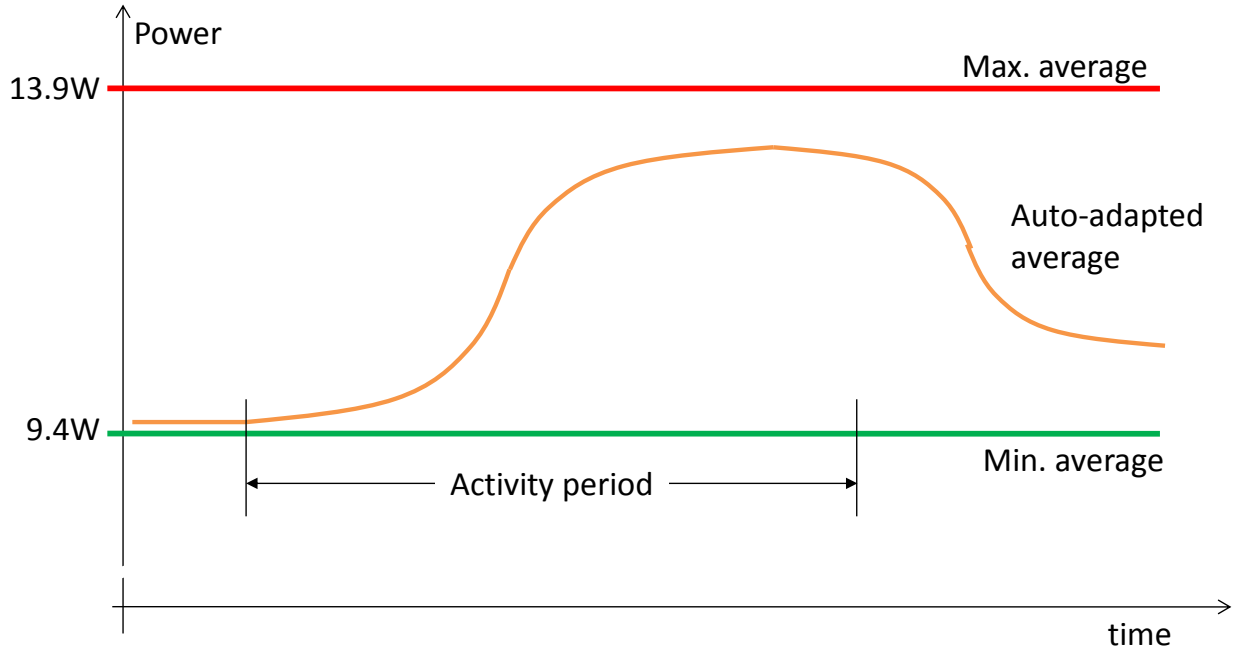


Figure 37: Average power consumption of "traffic light" home gateway.

Note that the absolute power values are less important for the result than the maximum span of 4.5W. The home gateways used for the demonstration are evaluation systems of LQDE, which are not optimized for power consumption. In a commercial system, the absolute power values will be lower, but the span of 4.5W can be applied to future commercial systems.

Applying the ECONET traffic mix leads to an average saving of 2.38W for the yellow home gateway. Over one year the savings accumulate to 20.31kWh. For the consumer this translates to about 5€ cost saving (assuming 25Ct/kWh), a value which might not be relevant for a private household. For the national economy, however the savings are important considering the high number of home gateways. For ten million households the savings accumulate to about 200,000 MWh. Taking this into account strengthens the argument of LQDE toward autonomic optimization.

7.6.2 Network connectivity proxy

The energy saving potential of the network connectivity proxy (NCP) can be potentially very high. For example, if a user keeps a PC running just to be able to receive messages during his/her absence, this could lead to additional power consumption between 50 and 100 W for several hours. With the home gateway always on, the additional consumption in the same time span is almost negligible (though it adds up in the long-run and over large numbers, but never to amounts comparable to the PCs).

The Network Connection Proxy (NCP) function carries on networking routines on behalf of client devices, which are temporarily unavailable for several reasons: they are not connected to the network (for example, mobile hosts out of network coverage), they are in low-power modes that hinder any activity, etc.

The principles of generality and flexibility are the main drivers behind the design and the development of the NCP, carried on by CNIT and INFOCOM in the scope of the ECONET project. The NCP is a portable software application that can fit different platforms and it can be sited in different devices within the home network (either networked devices, as, i.e., a Personal Computer, or networking devices as, e.g., a Home Gateway). The NCP is able to monitor the traffic passing through the covered network and identify specific traffic patterns, which require it to implement some actions. More in detail, the NCP can be configured to filter and manage specific traffic at different network layers: at the link layer (ARP packets), network layer (DHCP, ICMP echo-request packets), transport layer (new TCP/UDP connection requests, TCP keep-alive packets) and application layer (generic applicative heart-beating frameworks embedded in TCP/UDP flows). NCP actions consist of both answering packets addressed to the client devices it is covering and carrying out periodic tasks. Further operations supported by the NCP include waking-up sleeping hosts, buffering packets intended to unavailable clients, sending pre-defined packets, building packets at different layers of the networking stack The NCP provides a high-level interface to its clients for registering the requested behaviour and for notifying their availability (the full description of the interface can be found in D5.5). The NCP also works in the presence of Network Address Translation (NAT), which is the usual case for home gateways.

7.6.3 Function of the device in the network

The NCP is deployed at the edge of the ECONET network, within the customer site. Two sites were used in the demonstrator, to show two alternative deployments for the NCP (see Figure 38): on the home gateway (right side) or on another kind of low-power device (left side). A further host, called the remote peer, was implemented in the MLX server in the datacenter cloud (bottom of Figure 38).

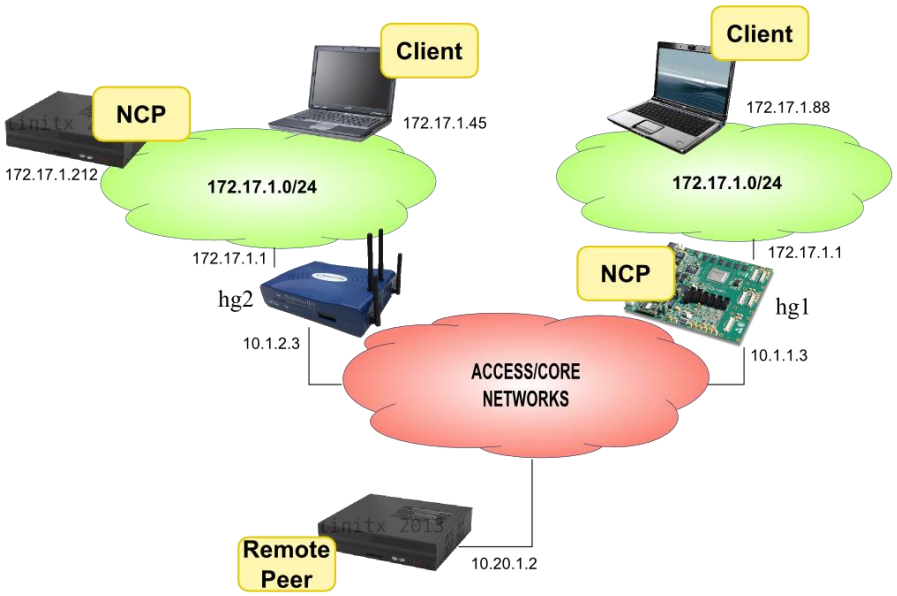


Figure 38. Layout of the customer sites for showing the NCP function in the ECONET demonstrator.

In the demonstrator, the NCP hides the standby state of its clients to other devices, by dealing with networking routines on their behalf. Energy saved by the NCP results from the difference between the power consumption of its clients in idle mode (when they are fully powered on to maintain network presence only) and in standby mode (technically, in suspend-to-RAM mode, which is a low power state that can quickly switch to full power on).

As briefly introduced in the previous paragraph, the NCP is able to perform several tasks on behalf of its client; however, for the sake of brevity, only the applicative heart-beating function was taken into consideration during the demonstration. Heart-beating is the most challenging function for the NCP when coupled with TCP, i.e., when packets are exchanged with a remote peer over a TCP connection, by hiding the transitions between the local device and its NCP; that means the TCP session is never broken and the remote peer is totally unaware of the state transition of the local device.

The target application for the demonstrator is a chat program, where clients connect to a common server and they are notified about the presence of other users. Chat clients run on terminals deployed at the customer sites, which are NCP clients too, while the chat server runs on the remote peer (see Figure 38). Without the NCP function the user must keep its laptop on to appear "on-line" to other users and to maintain its presence in the "chat rooms" it entered. Instead, when the NCP function is present, the client device can register the heart-beating message. It has to exchange messages periodically with the server to confirm its aliveness; once the device is put into standby, the NCP takes over its TCP session with the chat server, sends/answers periodic heart-beating messages and wakes up the device when new data are seen on that connection.

7.6.4 Type of technology used

The simplified NCP architecture is shown in Figure 39. The main components involved are packet filtering, packet processing and the communication protocol between the NCP and its clients.

Packet filtering is currently based on the pcap library. This library provides a common interface to get network packets on different platforms. Packet filtering considers both the packet protocol (ARP, DHCP, ICMP, TCP, UDP, …) and specific parameters carried in the packet header (e.g., MAC or IP addresses, TCP or UDP port numbers, TCP flags) and data (bit patterns). Packets are filtered to select the ones that need to be handled by the NCP and to select the right rule to apply.

Packet processing is the set of operations carried out by the NCP. Dealing with packets intended to other hosts breaks the common assumptions the networking stack of an Operating System is built upon, and forces to make extensive usage of raw sockets; raw sockets are programming interfaces that let applications the freedom to handle network protocol headers by themselves, thus allowing the transmission and reception of packets on behalf of covered devices. Another important task of the NCP is the "wake-up" of clients in standby mode; to this aim, the well-known Magic Packet technology (aka Wake-on-LAN) is used.

The communication protocol between client devices and the NCP should provide all the flexibility to request different kinds of actions, with different parameters and data. Yet, it should support autonomic and zero-configuration discovery of the NCP service and retrieval of the operations it supports. Since the customer networks are the main target for the demonstration, the current NCP implementation relies on the UPnP technology for this purpose. In the demonstration, the NCP exposes the UPnP NCP service designed in the ECONET project (see D 5.5).

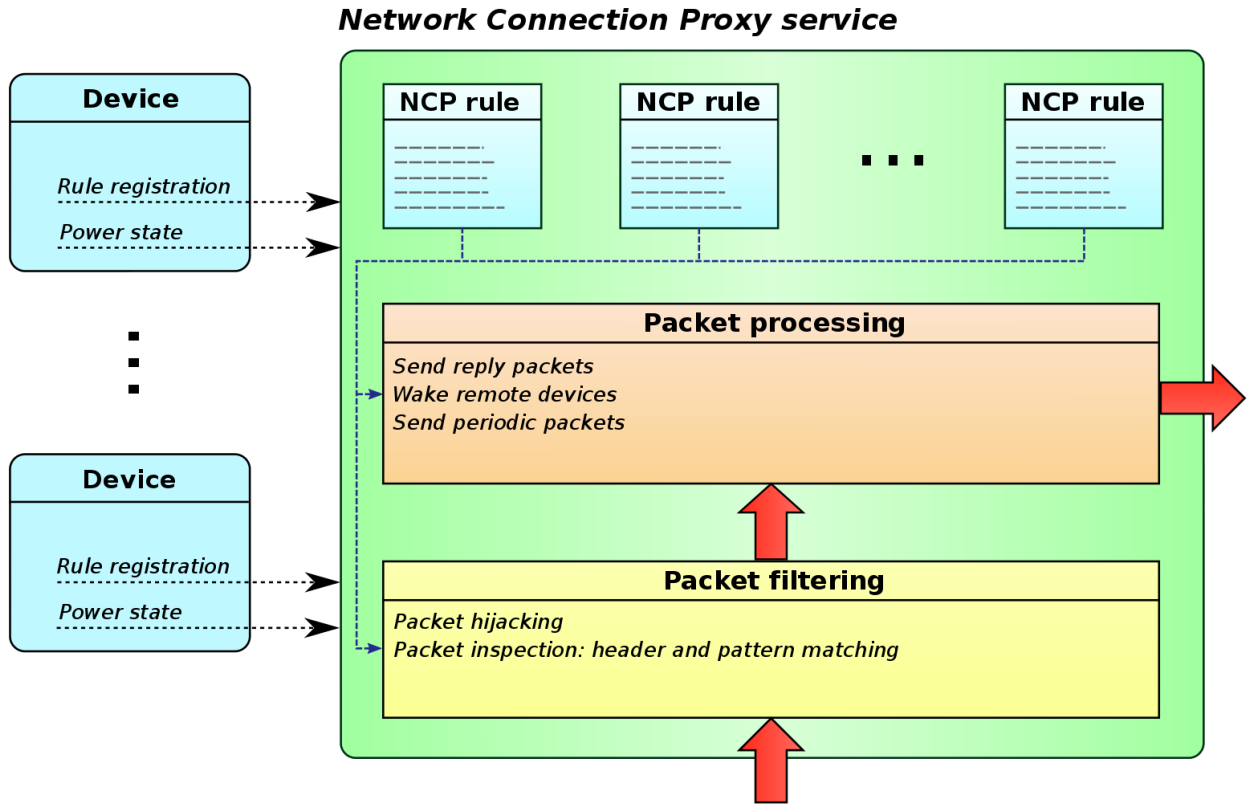


Figure 39. A simplified representation of the NCP architecture. On the left side, a communication protocol is used by clients to register the desired actions when they are unavailable and to notify their power state. On the bottom part, packet filtering diverts traffic intended for sleeping clients towards the NCP and inspects packets to select the proper action to apply. In the middle of the picture, packet processing includes different actions the NCP can perform: building/sending packets on behalf of sleeping devices, waking up hosts, buffering packets, and so on. NCP rules shown on top of the NCP service are an abstract representation of the behaviour requested by clients. There are several kinds of rules, corresponding to different actions that can be carried out by the NCP; each rule includes both the indication of the kind of packets and parameters to be matched to trigger the action and all data needed to perform the specific operation.

7.6.5 Characteristics of the device with respect to both transmission and energy aspects

The NCP service could run on different kinds of devices. Clearly, to achieve significant saving, the power consumption of the device hosting the NCP should be marginal with respect to the managed devices. Deploying this service on the home gateway is therefore an optimal solution, since this device is expected to be always on in most customer sites.

Another important aspect is the additional power drawn by the device when the NCP runs. Table 10 reports the power consumption of the LANTIQ home gateway deployed in the demonstrator under different conditions, including the presence of client devices in the site and the presence of the NCP running on the device. The figures in the "power drawn" column clearly show the NCP brings a very low overhead on the energy consumed by the home gateway. Table 10 only shows preliminary results, which should be enriched with more working conditions and measurements over longer time periods; however, note that power drawn by the home gateway is on the order of magnitude of power drawn

by PCs in the standby states, and this figure is at least one order of magnitude less than that in the idle state. Furthermore, Table 10 shows that the additional HG power consumption due to the support of the NCP (a process running on the NCP microprocessor) is negligible as it is lower than 0.5 W, i.e. less than 6% of the minimum HG basic power consumption when in the "green" mode. The power consumption gain provided by the home gateway power management daemon designed by LANTIQ (Section 7.6.1) is to be considered on top of the figures shown in Table 10..

Table 10. Measurement of HG's power consumption under different test conditions.

Condition	Power drawn
"Red HG"/No Home Host connected/No NCP	13.2W
"Red HG"/Home Hosts connected and working/No NCP	13,92 W
"Red HG"/Home Hosts connected and working/NCP Active	13,97 W
"Red HG"/Home Hosts connected, one sleeping/NCP Active	13,46 W
"Red HG"/Home Hosts connected, one sleeping/NCP Active and managing continuous Pings towards sleeping Home Host	13,49 W

7.6.6 What type of connections and relations with the other parts of the chain

One instance of the NCP function runs on the home gateway provided by **LANTIQ**[2].

The home gateways are connected to the **Ethernity DSLAM** by two VDSL links, according to the general picture shown in Figure 1. The remote peer at the bottom of Figure 38 is the MLX server. Traffic generated by each customer site is forwarded by its home gateway through the VDSL link and routed by CNIT routers towards either the other site or the data centre cloud, according to the actual destination.

7.6.7 Metrics and parameters collected during the demonstration

During the demonstration, both functional tests and measurements were carried out. Functional analysis showed that the NCP works as expected for all kinds of actions it implements; in particular, it can:

- be discovered automatically by clients and register the action requested for each of them;
- divert traffic back intended for sleeping clients towards the device hosting the NCP instance and back by means of ARP spoofing;
- answering echo-requests generated by the "ping" application;
- renewing DHCP leases;
- wake-up sleeping devices when a new TCP/UDP request is seen on a given port;
- answer TCP keep-alive messages, wake-up the host when new traffic on a TCP connection arrives and buffer packets until the host wakes up;
- generate heart-beating messages in UDP packets or over established TCP connections;
- migrate TCP connections with client devices before and after taking over their TCP sessions.

[2] One instance of the NCP runs on a different device than the home gateway just to show the possibility of deploying the NCP anywhere in the local network.

Metrics considered to evaluate NCP operation are:

1. the time taken to perform the required action;
2. the time taken to answer a new connection request towards a sleeping host;
3. the network overhead introduced by the UPnP signalling.

We did not collect the huge number of measurements that would be needed for a statistical evidence of the validity of the results, because of the time required to repeat every experiment. However, we carried out enough trials to derive the following conclusions:

- the latency taken by NCP operation is negligible.
- thanks to the buffering capability of the NCP, the latency on the establishment of new connections with a sleeping host is limited to the time to switch from standby to full power; for recent devices such time is about 3-4 seconds, and does not affect significantly the user experience (for example, when opening an SSH connection towards a sleeping device).
- the UPnP protocol carries on average more than one hundred bytes per packet (it uses XML descriptions) and sends more than one packet per second; however, on 100 Mbps - 1 Gps LANs the mean overhead is very limited (around 2 kbps).

8 Conclusions

This document has described the main characteristics, architectural organization and setup of the ECONET final demonstrator that was realized to highlight the experimental evidence of the results achieved by the project.

The demonstration was conceived to reproduce a complete TLC network, from the core devices to the subscriber terminals, in a configuration compatible with the asset of a medium-scale Telecom Operator covering both residential and enterprise network accesses, with the inclusion of datacenter facilities. It integrates quite a few of the green mechanisms proposed in Work Packages 3 and 5, and encompasses different equipment in terms of manufacturer technologies, interfaces, protocols and dimensions of the prototypes developed.

The demonstrator includes 15 physical prototypes of network devices, emulated networks, and a number of hosts and traffic generators, covering six different network segments: home, access, metro, data-centre, core, and transport. All these parts are coordinated and managed by another essential element of the demonstrator: the Monitoring and OAM structure.

The aim of the Demonstrator has been essentially twofold: on one hand, it represents a typical networked instantiation of the prototypes and solutions developed in the project; on the other hand, it is also an invaluable instrument to obtain real and significant measurements for the experimental assessment of the integration of the prototypes and technical solutions, and to verify the attainment of project goals in terms of energy efficiency.

The deliverable has described the demonstrator, its structure and topology, the test strategy and the single segments that compose the test bed. The main Sections described the topology of the test bed, the traffic profiles generated in the experiments, the functional architecture, the parameter and measurement collection methodology, and the demonstrator layout, along with the detailed description of the individual parts. The latter are structured in 6 “clouds”: i) Optical Transport, ii) Core, iii) Metro transport, iv) Data centre, v) Access, vi) Home network, plus an additional cloud for Monitoring and OAM.

The aim of the present deliverable was to present the detailed description of the demonstrator. Result of the experimental activity conducted on it will be the subject of D6.5 [7].

References

[1] The ECONET project, "Single device integration & demonstration," Deliverable 6.2, available online www.econet-project.eu.

[2] The ECONET project, "Control protocol extensions," Deliverable 5.5, available online www.econet-project.eu.

[3] The ECONET project, "Abstraction layer final definition", Deliverable 4.3, available online www.econet-project.eu.

[4] The ECONET project, "Final design of green technology for network device data plane", Deliverable 3.3, available online www.econet-project.eu.

[5] The ECONET project, "Green strategies at the control plane", Deliverable 5.3, available online www.econet-project.eu.

[6] The ECONET project, "Preliminary definition of green strategies at the control plane", Deliverable 5.1, available online www.econet-project.eu.

[7] The ECONET project, "Benchmarking and performance evaluation results", Deliverable 6.5.

[8] The Mellanox's Messaging Accelerator (VMA), https://code.google.com/p/libvma/

[9] R. Bolla, R. Bruschi, O. M. Jaramillo Ortiz, P. Lago, "The Energy Consumption of TCP," Proc. of 3rd ACM/IEEE Internat. Conf. on Future Energy Systems (e-Energy 2013), Berkeley, CA, USA, May 2013.

[10] R. Bolla, R. Bruschi, "An Open-Source Platform for Distributed Linux Software Routers," Computer Communications (COMCOM), Elsevier, vol. 36, no. 4, pp. 396-410, Feb. 2013. DROP source code available at https://svn.econet-project.eu/svn/.

[11] The ECONET Consortium, "Test plant evaluation criteria and representative test case," Deliverable 2.2, available online www.econet-project.eu.

[12] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, D. Suino, C. Vassilakis, A. Zafeiropoulos, "Cutting the Energy Bills of Internet Service Providers and Telecoms through Power Management: an Impact Analysis ," Computer Networks (COMNET), Elsevier, vol. 56, no. 10, pp. 2320-2342, July 2012.

[13] The Intel "Data-Plane Development Kit," URL: http://www.dpdk.org.

[14] L. Rizzo, "Netmap: a novel framework for fast packet I/O," Proc. of the 2012 USENIX Conf., June 2012.

[15] K. McCloghrie, F. Kastenholz, "The Interfaces Group MIB," IETF Request For Comments (RFC) 2233, URL: http://tools.ietf.org/html/rfc2863.

[16] The Net-SNMP project, URL: http://www.net-snmp.org/.

[17] The ECONET project, "Design of energy measurement technologies", Deliverable 3.5, available online www.econet-project.eu.

[18] The ECONET project. "Network Connection Proxy: Maintaining Network Connectivity for Sleeping Hosts", Internal Report n.7 annexed to the Deliverable 3.3, available online www.econet-project.eu.

Appendix A: Example of topology XML

The example below describes a cloud with 3 devices:

- The cloud has a GAL interface where energy related information can be accessed by the management system through the REST API.
- TEI-1 switch has 3 modules, one backplane with a CPU and two line-cards with 2 ports each. Port 1.1 also has a status metric named “speed”.
- TEI-2 switch is similar, but it only contains a single line-card. One port is connected to a port on TEI-1, while the other is connected to an external device.
- TEI-4 is a smaller device with 3 integrated ports (i.e., no line-cards). One of these is connected externally, while the others are probably unconnected.

```
<?xml version="1.0" encoding="UTF-8"?>
<Topology xmlns="https://www.econet-project.eu/topology-schema"
   rootType=”cloud” top_id=”tei-cloud” name="Metro">
  <Gal_URL>http://econet.ericcson.com/tei-gal/</Gal_URL>
  <Device type="switch" id=”sw1”>
      <Name>TEI-1</Name>
      <Description>ECONET Switch No. 1</Description>
      <visualization>
            <position x="0" y="25"/>
      </visualization>
      <Module id=”sw1-mainboard”>
            <Name>Backplane</Name>
            <Component type=”CPU” id="sw1-CPU"/>
      <Module>
      <Module id=”sw1-m1”>
            <Name> Card-1</Name>
            <Component type=”port” id="sw1/1/1"/>
            <Component type=”port” id="sw1/1/2"/>
            <Component type=”fan” id="sw1/1/fan"/>
      </Module>
      <Module id=”sw1-m2”>
            <Name> Card-1</Name>
            <Component type=”port” id="sw1/2/1">
                  <StatusMetric name=”speed” type=”numeric” units=”Mbps”

      access=”snmp://15.24.32.155@publicc/1.3.6.3.4.2.3.1”/>
            </Component>
            <Component type=”port” id="sw1/2/2"/>
            <Component type=”fan” id="sw1/2/fan"/>
      </Module>
  </Device>
  <Device type="switch" id=”sw2”>
      <Name>TEI-2</Name>
      <visualization>
            <position x="0" y="45"/>
      </visualization>
      <Module id=”sw1-mainboard”>
            <Name>Backplane</Name>
            <Component type=”CPU” id="sw2-CPU"/>
      <Module>
      <Module id=”sw2-m1”>
```

```
            <Name> Card-1</Name>
            <Component type=”port” id=”sw2/1/1”>
                  <Connection peer="sw1/1/1" />
            </Component>
            <Component type=”port” id=”sw2/1/2”>
                  <Connection peer="MLX_CLOUD-dev1/1/1" />
            </Component>
      </Module>
  </Device>
  <Device type="switch" id=”sw4”>
      <Name>TEI-4</Name>
      <visualization>
            <position x="0" y="45"/>
      </visualization>
      <Component type=”CPU” id="sw3-CPU"/>
      <Component type=”port” id=”sw3/1”>
            <Description>Mgmt port on device mainboard</Description>
            <Connection peer="ALCATEL-router4/2/1" />
      </Component>
      <Component type=”port” id=”sw3/2”>
      </Component>
      <Component type=”port” id=”sw3/3”>
      </Component>
  </Device>
</Topology>
```

Appendix B: The NASK/WUT small-scale demonstrator

This appendix presents the design of energy-aware control system reducing power consumption cost of QoS provisioning in an experimental network of Linux Software (SW) routers. The system, developed by WUT and NASK, automatically adjusts performance of the network to the workload generated by the network traffic.

B.1 Architecture of the testbed

An experimental network was built and set up to facilitate testing of the green technologies in practice. Instead of dedicated network appliances, PCs with routing capabilities were used to forward layer 3 traffic. The network consists of seven DELL Precision T1650 nodes, each one equipped with Intel Core i7-3770 CPU with 16GB DDR 1600MHz memory and Broadcom 5719 QP 1Gb Energy Efficient Ethernet 4-port network cards. Each CPU is in the network capable of operating at 16 frequencies, ranging from 1.6 to 3.401 GHz. Power consumption monitoring is supported by 1-phase electricity meter for active energy measurement with RS485 interface. Topology of the network corresponds to the core of WARMAN - Warsaw Metropolitan Area Network used by academia and research institutions. Each router can operate in two states: active and suspended. All communication ports can operate in one of three possible active states with the following throughput: 10Mbps, 100Mbps and 1Gbps.

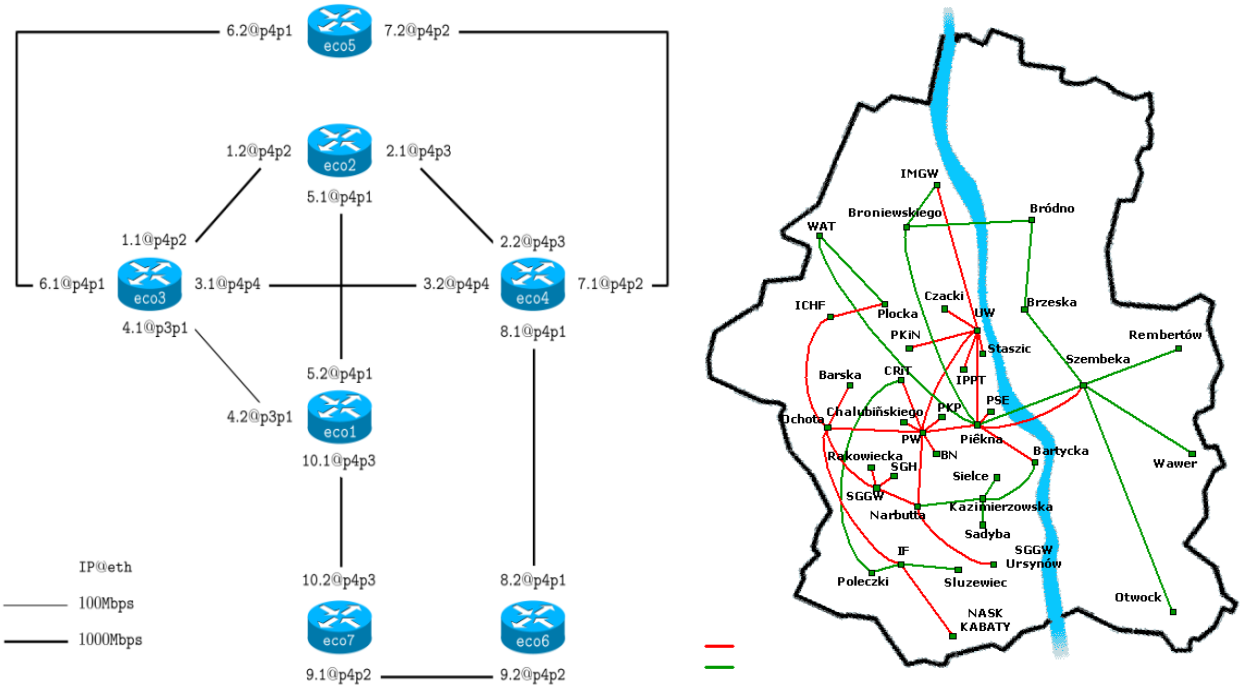


Figure 40. Topology of the testbed and WARMAN - Warsaw Metropolitan Area Network.

To control traffic flows in the network a port/policy-based source routing mechanism (PBR) was implemented. The mechanism implements functionality of the MPLS standard using the capabilities of the *iptables* filtering technique. In addition, each router was equipped with a *tcpdump*-based probe performing real-time packet inspection. Both mechanisms, when activated, generate CPU workload correlated to the amount of traffic forwarded by a router. Consequently, two types of nodes were distinguished in the network:

- packet analyzing nodes
- packet forwarding nodes.

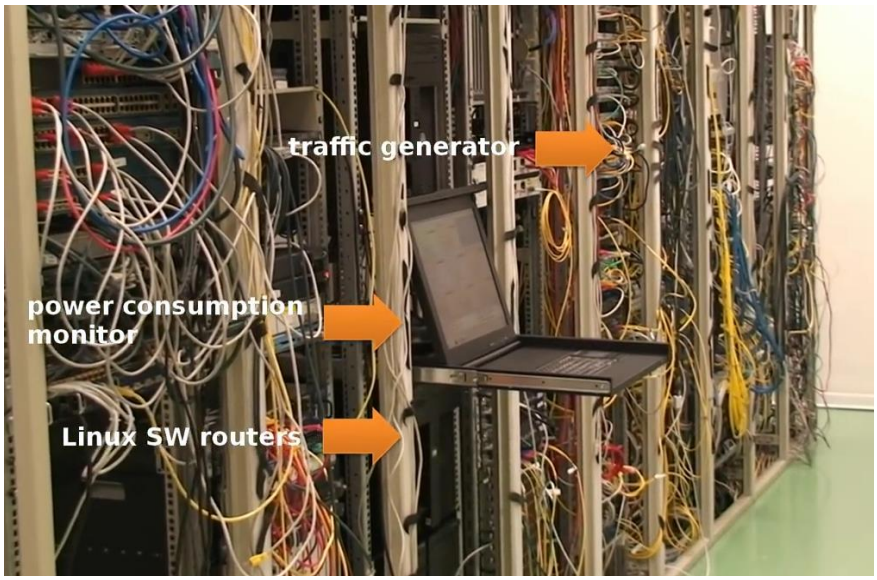


Figure 41. Testbed picture.

Each node type was subject to the performance profile identification procedure. As a result, a collection of best response functions $\hat{\mu}_\gamma$ was constructed, each function corresponding to a fixed value of weight γ assigned to the power consumption cost. A best-response function determines the CPU frequency at which an observed workload is served at minimal cost. Implemented as a lookup table for the *cpufreq* Linux kernel module it was used to control CPU frequency of each router in the network. Details of the control policy design are presented in Chapter 3 of D5.3 [5].

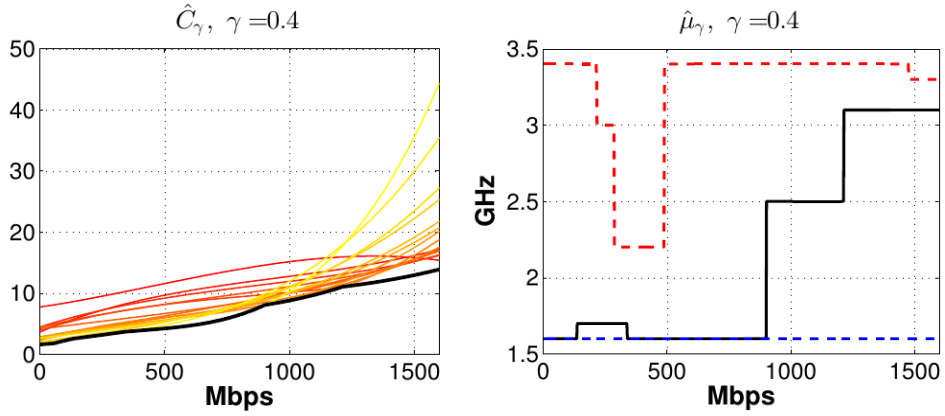


Figure 42. Examples of performance cost and best-response function of a traffic processing node (black).

B.2 Architecture of the control system

A two-layer structure was developed to control the performance of the experimental network. The structure consists of a network control and monitoring layer, operating at the level of network manager, and a local control and monitoring layer, operating at the level of network devices.

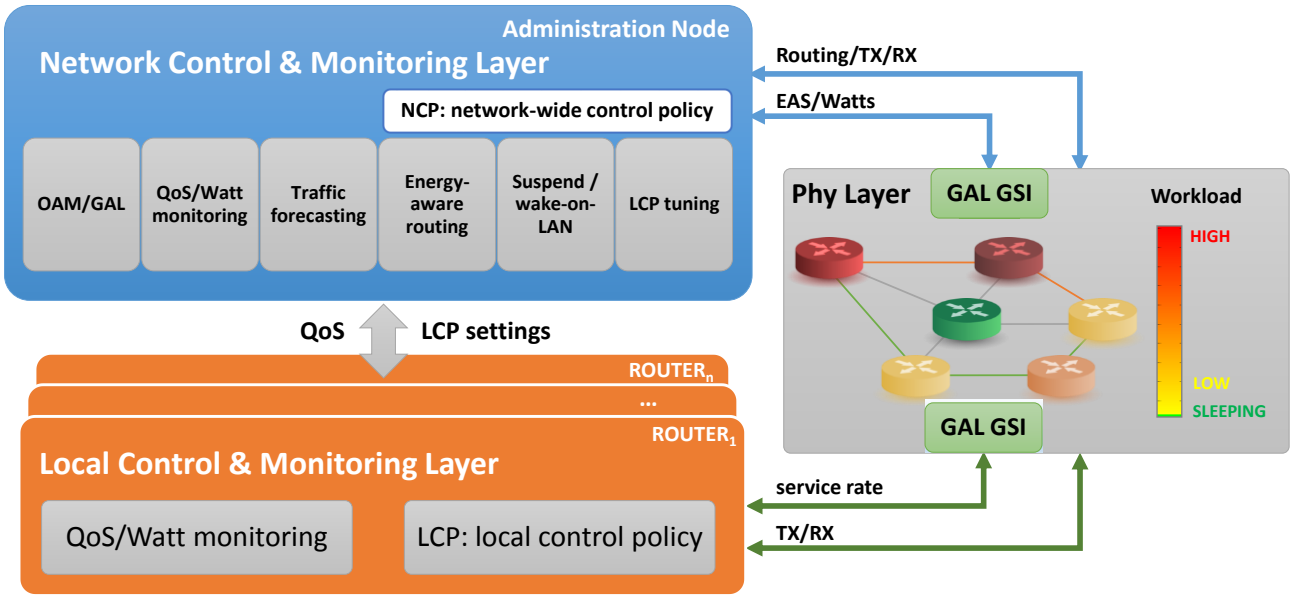


Figure 43. Architecture of the developed control system.

The upper layer is responsible for periodical optimization of energy-aware routing and energy-saving configuration of the network. The lower layer executes commands received from the upper control layer, collects measurements and controls routing performance. Every sampling period the NCM layer collects measurements of the network traffic, power consumption and QoS. The collected measurements are given visual representation and are used to generate peak-workload forecasts. The forecasts, together with the performance profiles submitted by the routers, provide an input for energy-aware optimization of network configuration. In addition, in the period between the subsequent calls of optimization procedure the NCM layer performs regulatory control of operations performed by the LCM layer. The goal of the process is to maintain QoS requirements along the activated routing paths in the network. Details of the control loop design are presented in Chapter 4 of D5.3 [5].

The lower control layer has a direct access to each router in the network and is in charge of executing control commands. In the developed environment it is responsible for two operations:

- energy-aware CPU frequency control,
- router suspending and activating.

The layer is also responsible for collecting measurements of routing performance and power consumption.

The GUI of the layer illustrates measurements collected from the network, in particular its active topology, power consumption and CPU load of each router. Based on the collected measurements the upper control layer generates peak workload forecasts for each registered traffic stream. These

forecasts, together with the identified performance profiles of the routers, are used to optimize routing tables and to set up activity levels of the nodes.

To calculate routing tables and energy-saving configuration of the network a constrained optimization problem is solved. As an input the problem solver receives the power consumption profiles and the peak workload forecasts of traffic flows. As an output the optimization procedure returns a description of energy-efficient routing paths capable of forwarding the forecasted amount of traffic. Based on the obtained solution the routing tables are next generated together with the required configuration of the network. Reconfiguration of the network is commanded by the NCM layer and executed by the LCM layer. Whenever traffic workload is forecasted to be high, additional network resources are activated. In order to keep the required QoS level some of the connections may also be redirected to new routing paths. Whenever traffic intensity is forecasted to be low, the system automatically reduces activity level of the routers. As a result the connections are redirected to new routing paths.

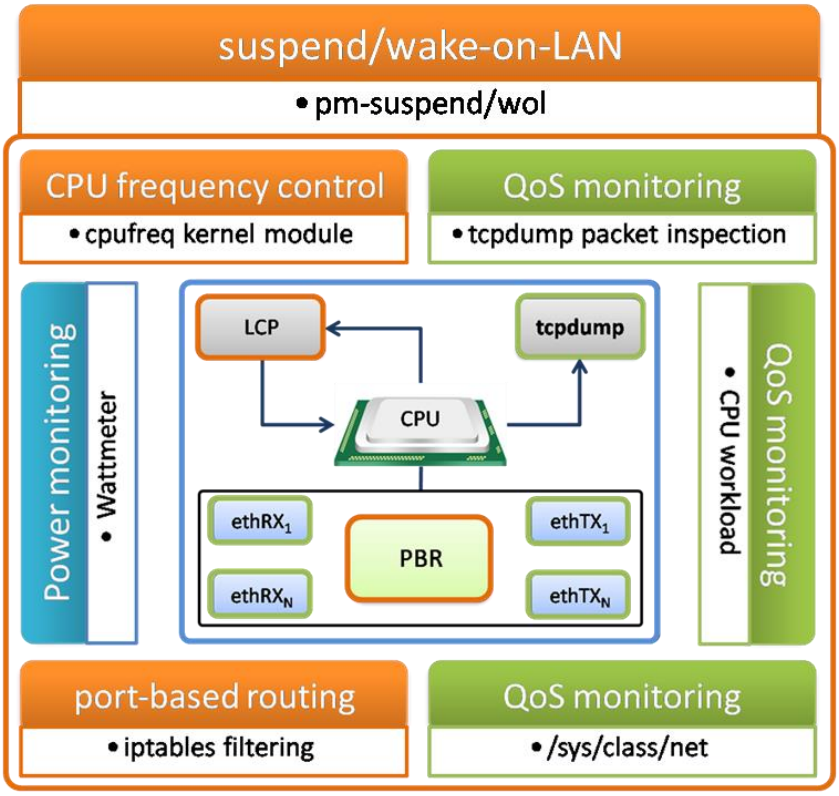


Figure 44. Linux SW router as control object.

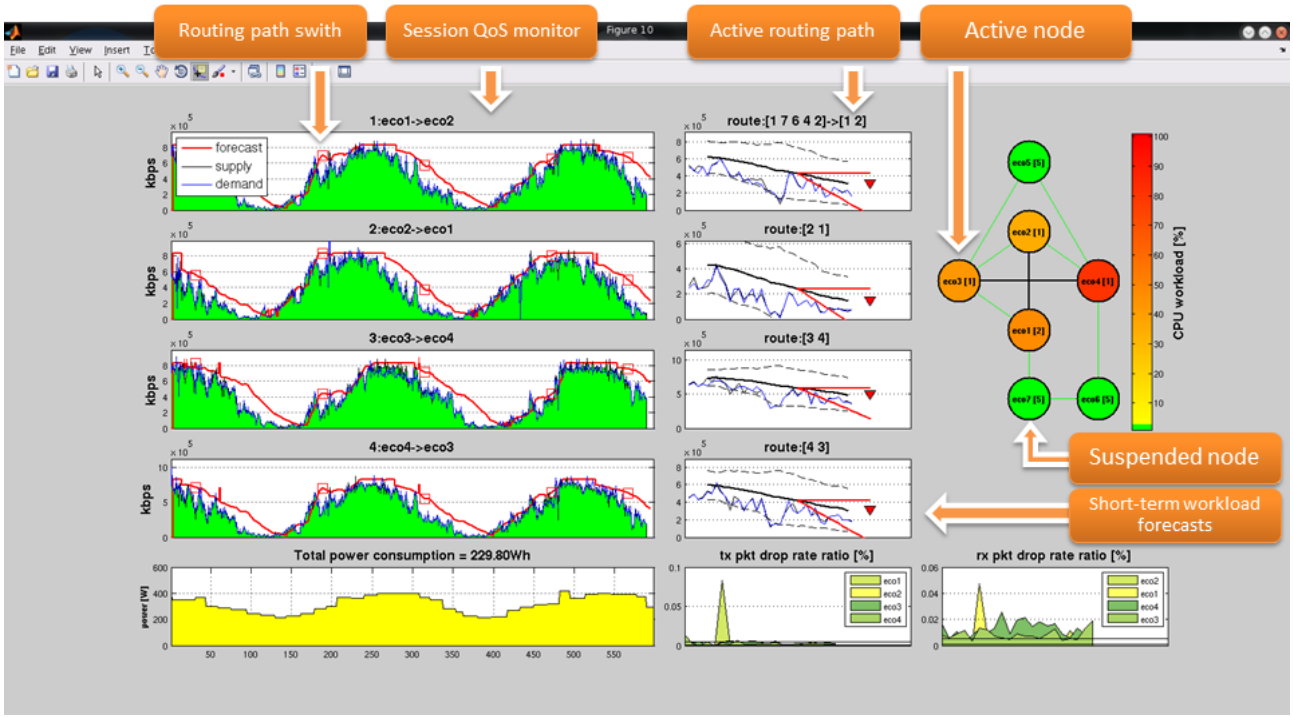


Figure 45. GUI of the upper control layer.

B.3 Results of experiments

The control system was forced to respond to the noisy periodic UDP traffic workload, observed over long control horizons (days/weeks), with the noise generated based on TCP/IP link traffic trace. The main goal of the experiments was to estimate the expected performance bounds of the energy-aware networking in the Linux-based environment built of the COTS components.

The results of experiments with the control structure described in Section 4.2.4.2 of D5.3 [5] are summarized below. To obtain a comparison of outcomes the average performance of each control strategy was divided by the worst-case average performance. The ratios are presented in the bar charts.

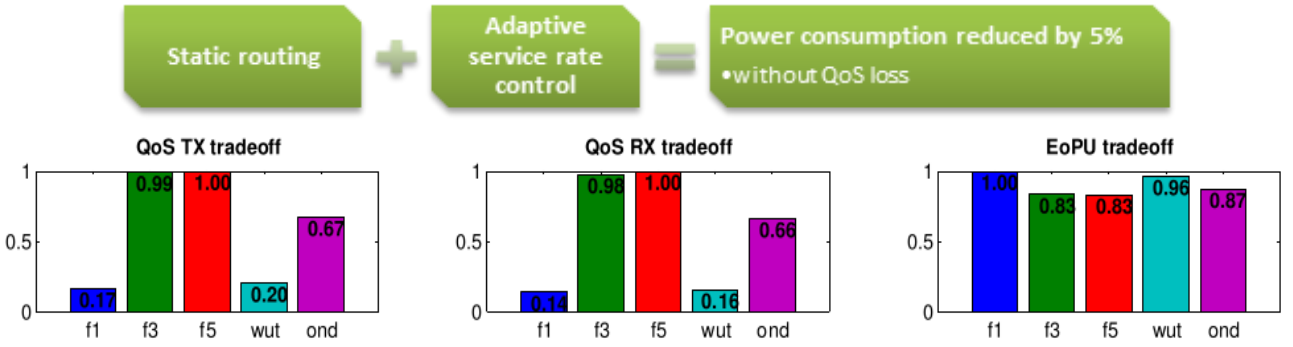


Figure 46. Comparison of local control policies.

The results show that in the addressed environment the applied CPU performance control mechanisms (LCPs) were able to reduce power consumption by 5% *without* QoS deterioration. The following LCPs were compared:

- **f1**, **f3**, **f5:** stationary control strategies defined by the best-response function $\hat{\mu}_\gamma$ (lower-bound for optimal control strategy) with weight γ equal to 0.25, 0.5 and 0.75, respectively;
- **wut**: adaptive control strategy based on $\hat{\mu}_\gamma$ function with γ dynamically adjusted to the observed packet drop rate by an external (feedback) process,
- **ond:** default Linux *ondemand* control strategy.

As can be seen, the best QoS index is obtained by strategy **f1**, keeping the maximal CPU frequency fixed. This, however, comes at the cost of higher power-consumption. The same holds for strategy **wut**, adjusting the aggressiveness of power-consumption to the observed level of packet drop. The **ond** governor reached a high efficiency in terms of both QoS and power-consumption indexes. However, it was outperformed in terms of efficiency of power-consumption (EoPU) by strategies **f3** and **f5** exploiting the specific operating characteristics of the system.

The results support several general and somewhat intuitive conclusions regarding the performance of CPU frequency control mechanisms in the Linux software router. First, a justified attempt to reduce power-consumption can be made if a sufficient supply of service rate is provided by the system. Second, the observed variations of power-consumption costs are low in comparison to the variations of QoS-costs. High power-efficiency results in low QoS, and vice-versa. Third, the results of experiments confirm the efficiency of the dynamic over-provisioning control policy. This is in accordance with the intuition that the processor should be able to accommodate a reasonable amount of fluctuations surpassing the expected demand level. Finally, the obtained outcomes support the following design guidelines. Whenever the weight assigned to power-consumption cost is low, i.e. the system is performance seeking, the maximal service rate should be selected. If the system is power-consumption neutral, i.e. the same weight is assigned to QoS cost and power-consumption cost, the best-response rule with a moderate over-provisioning level performs as well as the *ondemand* governor. In case of a power-consumption-averse system, assigning high value to low power-consumption guarantees, control rule based on the best-response function $\hat{\mu}_\gamma, \gamma = 1$, may be recommended.

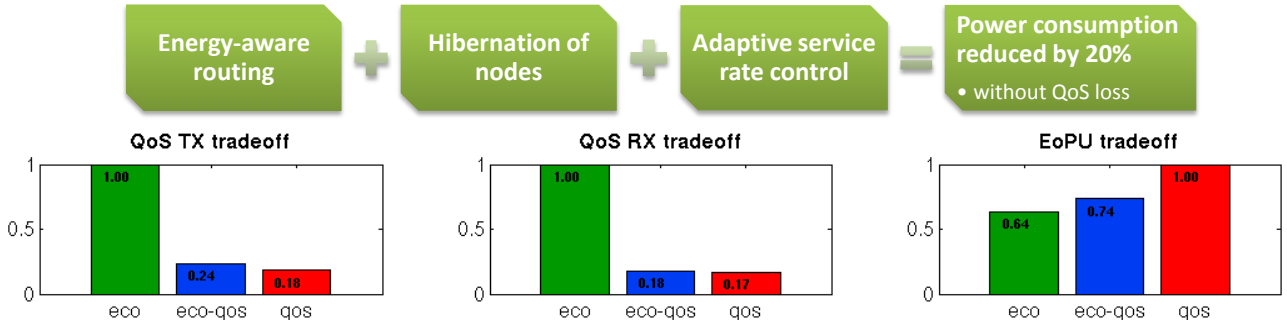


Figure 47. Comparison of network-wide control policies

Improved outcomes were obtained when traffic re-routing and hibernation of nodes were also applied. In this case, the system was able to reduce power consumption of the network by 20% keeping the QoS at high level. The following network-wide strategies were compared experimentally:

- **eco:**
  - NCP: energy-aware routing activating and suspending nodes on demand,
  - LCP: stationary CPU frequency control strategy ($\hat{\mu}_\gamma,\ \gamma = 1$).
- **eco-qos:**
  - NCP: energy-aware routing activating and suspending nodes on demand,
  - LCP: adaptive CPU frequency control strategy (wut);
- **qos:**
  - NCP: static routing.
  - LCP: stationary CPU frequency control strategy (f1).

Additional experiments were also performed to test the resilience of the control framework to the quality of forecasts of demands. Two series of experiments were carried out to validate and compare the effectiveness of two optimization schemes, **LNPb** and **LNHP,** discussed D5.3 [5]. The testbed was used to transmit 10 flows meant to carry ~100 Mbps of offered load each. The experiments comprised of two scenarios:

- (F1) the real traffic was equal to the forecast,
- (F2) the real traffic was 50% higher than the forecast.

The summary of experiments performed with the **cNCP** control structure described in Section 4.2.4.1 of D5.3 [5] is presented in the tables below. They collect the values of total traffic measured on network interfaces, power consumed by whole network, percentage of power saved and power effectiveness coefficient.

Table 11. Results of testbed experiments using the LNPb solution.

Scenario	Measured traffic [Mb/s]	Power consumption [W]	Power reduction %	Power effectiveness [W/Mb/s]
F1	999.4	202.5	26	0.203
F2	734.2	197.1	28	0.268

Table 12. Results of testbed experiments using the LNHP solution.

Scenario	Measured traffic [Mb/s]	Power consumption [W]	Power reduction %	Power effectiveness [W/Mb/s]
F1	1004.7	208.3	24	0.207
F2	686.2	200.3	27	0.292

The reference solution for results presented in the above tables is typical shortest path routing with all nodes switched on. The power consumption in this case is 274 W. The collected data show that solutions calculated for underestimating forecasts (F2) cannot provide enough bandwidth to cover all demands fully. It must be noted, however, that such reduction of traffic results in reduction of power consumption too, as the overall load of the network is lower. To simplify comparison the effectiveness

coefficient was introduced. It shows how much power is used by the network to transfer 1 Mbps of traffic. With this coefficient it is clearly visible that although the power consumption is lower in F2 scenario the effectiveness of such a network is significantly worse, i.e., that if a certain amount of data is to be transmitted it will take longer time and consume more power in F2 case.

The difference between exact and heuristic solutions for the F1 scenario results from inaccuracies in equipment model, namely it can be observed that the physical rate limit of the interface in 100 Mbps mode is 95.7 Mbps. The solution found by LNPb method switches two links into this state providing power savings over LNHP. However, in the real network two flows instead of one are slightly restricted. Another deviation in traffic rates, i.e., 101 Mbps instead of 100 Mbps results from simplifications in traffic generation and measurements made on 1 Gbps links.

Table 13. Comparison of flow rates (LNPb and LNHP solutions; testbed network).

	LNPb method		*LNHP* method	
	Scenario		Scenario	
Flow	F1	F2	F1	F2
1	101.0	101.0	101.0	101.0
2	101.0	47.8	101.0	47.9
3	101.0	101.0	101.0	101.0
4	95.7	47.9	101.0	47.9
5	101.0	48.1	95.7	48.7
6	101.0	95.7	101.0	47.1
7	95.7	48.0	101.0	47.0
8	101.0	95.7	101.0	95.7
9	101.0	101.0	101.0	101.0
10	101.0	48.0	101.0	48.9