Evaluation of a Policy-Based Network Management System for Energy-Efficiency

Guilherme C. Januário, Carlos H. A. Costa,
Marcelo C. Amaral, Ana C. Riekstin, Tereza C. M. B. Carvalho
Escola Politécnica, University of São Paulo
São Paulo, Brazil
{gcjanuario,chancosta,marceloamaral,
carolina.riekstin,terezacarvalho}@usp.br

Catalin Meirosu
Ericsson Research, Packet Technologies,
Sockholm, Sweden
catalin.meirosu@ericsson.com

***Abstract*—Energy efficiency features are being integrated in network protocols and management systems. Simulations can provide input on how a particular algorithm would perform in different network conditions. However, building an environment that is able to comprehensively account for interactions between different network functions is difficult. A combination of emulation and implementation of major energy efficiency features provides a view closer to what may happen in a real deployment. Previously, we described a policy-based network management system that optimizes network paths with respect to energy efficiency. In this paper, we evaluate the system using software routers for emulating network equipment functionality. We discuss extensions we developed and trade-offs of employing a state-of-the-art Linux distribution to emulate an energy efficient router. The article then presents results from the evaluation of the management system and argues about the benefits of the emulation environment.**

Keywords*—*Network management; Energy savings; Sustainability; Testbed.

I. Introduction

The Global e-Sustainability Initiative (GeSI) reported that the Information and Communication Technology (ICT) sector was responsible in average for 2% of global carbon emissions in 2008. Such an amount is expected to grow 6% per year until 2020 [1]. Networks are responsible for 70% of the amount [2]. Telecommunications service providers usually over-provision link bandwidth and use redundant equipment to assure the service level agreements (SLA) [3]. Backbone links are typically used much below their capacity [2]. Operating nodes at full capacity all the time, regardless of the traffic demand and distribution, leads to energy inefficiency in the network infrastructure. Opportunities thus exist to reduce energy consumption.

Bolla et al. [4] identified three categories of techniques for reducing energy-consumption in networks: re-engineering, dynamic adaptation, and sleeping/standby. The first category refers to energy efficient design, materials, and components. The second category refers to adapting the network devices according to current traffic load and service requirements. The third category refers to turning off underutilized subsystems of devices. The last category requires that the network devices support at least one power saving mode and be able to exchange and interpret specific control messages, such as requests to put a device to sleep.

To tackle these issues a holistic sustainability test environment and an energy-efficient network management system with centralized decision are necessary. Centralized sustainable architecture is necessary when not all the information for evaluating power consumption, network availability, and traffic flowing is available locally. The system must also coordinate energy-efficiency features in a heterogeneous network. A heterogeneous network possibly makes use of legacy equipment that lacks support to energy-efficiency features. Management point of view is provided once the system equips the manager with policies and enables analysis of the trade-off between energy savings and level of service degradation.

A test environment for evaluating the features of an energy efficiency oriented network requires that the routers support different power states and provide energy consumption information. Real-time and fine-grained power consumption reports are rarely found in currently available devices. For allowing such a real-time measurement, power models can serve as a way to parameterize power consumption. We consider linear power profiles, determined by the consumption when a device handles no load, along with a scaling factor of load increase.

Power profiles can be used for emulating approaches to evaluate envisioned features. In such a case, software routers are commonly used. When using software routers the emulation of device capacity relies on traffic shaping to realistically emulate an overloaded device and the related dropped packets. Because default routing protocols do not necessarily define the optimal path with respect to energy efficiency and QoS constraints, traffic engineering is a way of ensuring the usage of such paths. Figure 2 summarizes the test environment requirements described above.

We previously presented an architecture that fulfills the discussed requirements, in [5]. The policy-driven energy-efficient NMS proposed therein, SustNMS, takes into account the assurance of QoS and ensures fast response to either failure or sudden traffic increase. For each decision about in which mode a node should be (e.g., power-saving/normal mode), an online analysis of reliability provides further clarification on the potential consequences of the decision [6].

The present work details aspects of SustNMS implementation and explains challenges on how to set up a test environment for validating energy-efficiency management systems, discussing the results. We developed an integrated testbed

978-3-901882-50-0

to evaluate power-aware network management solutions by emulating power saving states and supporting traffic steering.

The remaining part of this paper is organized as follows. Section II describes related works. Section III presents the SustNMS architecture and its implementation details. Section IV presents a novel approach for testing energy network management systems and the necessary steps in doing so. Section V discusses the experimental results. Section VI brings a discussion, and Section VII concludes this work.

II. Related Works

The Internet Engineering Task Force (IETF) standardizes power-related data from network equipment. The Energy and Power Management Information Base draft provides a model for energy management information support [7]. Approaches like the Energy Management System [8] propose management systems that require awareness of the entire network. The authors suggest the management of powering on/off network elements by providing embedded IT services in devices, e.g., Web Services. The solution then turns off vacant links and, if possible, routers. Chaudhari et al. [9] propose an energy aware network management system based on an energy-aware MIB and on a centralized decision management system. The authors propose to add a new state to the MIB portion defined in RFC 2863, namely the "sleep mode". Such solution could be implemented when all network routers support this state and are SNMP-enabled.

A centralized traffic engineering mechanism was suggested in [3] for route calculation using the network topology and the traffic matrix. The objective is to maximize the energy efficiency by putting line-cards to sleep while keeping network performance at desired levels. The MiDORi (Multi-layer, path, and resources Dynamically Optimized Routing) project [10] proposes a Path Computation Engine for calculating optimal energy topologies that can accommodate all traffic.

A test environment in which the above-mentioned solutions could be validated and compared is as important as the solutions themselves. Environments for experimental network research, such as the ns-2 simulator and the Emulab testbed, are well known in the research community. A testbed that enhances Linux-based switches so they support diverse experiments on green networking protocols and configurations was proposed by Chabarek and Barford [11].

III. System architecture and implementation

This section gives an architecture overview and describes dynamics and implementation details of SustNMS.

A. Architecture overview

The architecture of SustNMS is depicted in figure 1. The system requires that the devices have some features enabled (further details on Section IV-A). SustNMS extends the policy-based network management architecture defined by IETF by including three modules: the Model Repository, the Quality of Service Monitor, and the Sustainability Monitor.

The first module included in SustNMS architecture, the Model Repository, comprises two submodules. One submodule stores power consumption models, which are static parameters that define power profiles. The other submodule stores availability models. Availability models are failure and repair rate information. The second included module, the Quality of Service Monitor, includes two submodules. One submodule dynamically evaluates network availability every time a new configuration is calculated by the undermentioned PMF. The other submodule dynamically collects performance indicators of each node of the network.

The third included module, the Sustainability Monitor, is composed by a single submodule, the Energy Efficiency Evaluator (EEE). The EEE evaluates the instant power consumption of each node, thus allowing an energy efficiency evaluation of the entire network. The Device Updater (DU) is the module responsible for collecting data and for applying changes to each network device. For doing so, the DU communicates with each instance of Policy Enforcement Point (PEP). PEPs are responsible for executing the necessary actions that make the network comply with the management policies defined by the network manager. PEPs may perform device-specific condition check and validation, if necessary.

The architecture defined by IETF is composed by the Policy Management Framework (PMF), which comprises other three submodules, to wit: 1) the Policy Management Tool, an editing and validating tool, 2) the Policy Decision Point (PDP), a module made of trigger detection and handling, and 3) the Policy Repository, a storage [5].

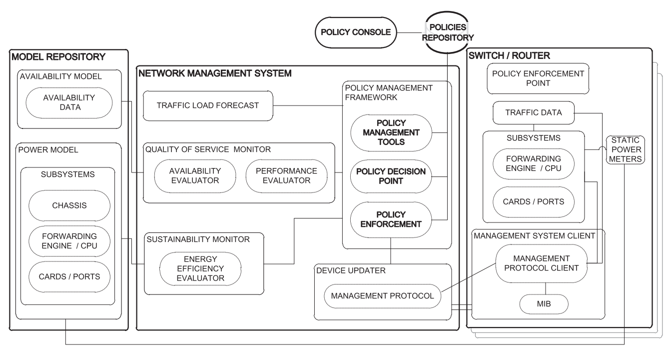

Fig. 1. SustNMS Architecture [5]

B. Implementation Details

In order to evaluate the proposed system, the architecture presented in Section III-A was implemented in Python. SustNMS runs on top of an MPLS data plane network. MPLS is widely used in commercial networks [12]. The possible paths defined by MPLS are set beforehand, therefore no path calculation is necessary during network operation. As pointed out by Dongmei et. al. [13], using backup paths with MPLS provides fast rerouting and low overhead in the control plane.

The starts loading the PDP, a module attached to a single DU and a single EEE. The regular actions of the PDP are waking the DU and triggering the EEE after the internal representation of information from all devices has been updated. Static information such as the power profile and the availability model of each device is stored within the system (on the Model Repository). Time-varying information, e.g., device traffic load or power state, changes dynamically according to the demand of network clients. The DU accesses such data via SNMP.

After the probing phase, the Quality of Service Monitor evaluates availability and performance of the network, allowing the PDP to check whether policy constraints remain satisfied when applying Algorithm 1. The Policy Repository and the Availability and Power Model Repositories are stored in a MySQL database.

Algorithm 1: Defining which predefined tunnels should be working

input : network topology and a set of existing flows
output: C_{min}, a set of paths that must be set up

R denotes a router
C is a set of paths, denoting a candidate solution
$S0 \leftarrow$ *set of sets that grant connectivity*
$S1 \leftarrow$ *set of paths that grant flow 1*
$S2 \leftarrow$ *set of paths that grant flow 2*
$CS \leftarrow \{distinct\ C | C \in A \cup B \cup D,$
$\forall (A, B, D), A \in S0, B \in S1, D \in S2\}$

foreach C *in* CS **do**
    if C *satisfies policy* **then**
        foreach R *in* CS **do**
            $l_R \leftarrow$ *load implied by solution*;
            $P_C \leftarrow P_C + Consumption(l_R, R)$;
        if $P_C < P_{min}$ **then**
            $P_{min} \leftarrow P_C$;
            $C_{min} \leftarrow C$;

return Cmin

IV. Test Environment

The present work extends existing approaches to perform testbed-based experiments on power-aware networks. The extension consists in adding, by means of network management tools, the support to the evaluation of costs and benefits related to the coordination of power consumption optimization. This way, one can analyze potential trade-offs between saving energy and degrading performance in future algorithms and hardware. The objective is to define a test environment that allows the evaluation of power-awareness techniques from a management point of view.

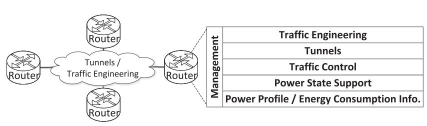

Fig. 2. Test environment for evaluating features of an energy-efficiency oriented network

A. Development of the Test Environment for SustNMS evaluation

The present subsection describes, as a proof of concept, an implementation of the afore-mentioned general test environment.

The simulation environment for evaluating SustNMS is based on Linux software routers. Usage of Linux-based routers is growing [11]. For this reason, the described test environment is applicable to a large set of devices. For emulating different routers, the network is built with virtual machines, using VMWare vSphere (ESXi). The virtual machines were interconnected through VMWare vSwitch (Layer 2 forwarding, VLAN tagging). Disabling the network interface cards (NICs) related to the data plane emulates the standby state. Control plane NICs are never disabled, so to keep network presence of the corresponding router. In addition to disabling data plane NICs, the system relies on power profiles to model the energy consumption of standby modes. Standby is a mode in which the device is not handling any load, but from which it is capable of waking up after an elapsed state switching delay. Such delays are stored on the Availability Model of SustNMS.

The presented test environment for evaluating power-aware management systems shows how QoS degrades whenever topology changes, that is, whenever some device gets in/off standby. SustNMS performs the necessary calculations by taking into account the realistic time taken by the device when becoming fully operant. This task is implemented based on [6].

For traffic shaping, the solution is relying on the Linux Traffic Control (LTC), a kernel-based feature. Such a feature allows one to control rate limits and to manage bandwidth. By having the complete control over the packets that traverse the device, LTC makes it possible to manage the outgoing traffic of available NICs. The adopted way of using LTC is relying on the Hierarchical Token Bucket queuing discipline, one of the most common disciplines for traffic shaping [14]. Such a scheduling policy allows a realistic modeling of the policies that are implemented in physical devices.

Regarding traffic engineering, a software-based approach, the MPLS-Linux project [15], is applied to support MPLS. The used operating system is Linux Debian, kernel 2.6.27.24. The resulting environment supports Ethernet and PPP interfaces, virtual MPLS tunnels, label stacking, recursive lookups, and DiffServ. Using MPLS assures hard QoS requirements, such as bandwidth, link attributes, explicit route, backups, and fast reroute. In SustNMS, a server infers information about the tunnels by using SNMP to access RFC1213-MIB variables (to wit, $ifInOctets$ and $ifOutOctets$). By using CLI commands through SSH, the server defines which tunnels the ingress routers should configure for the incoming flows. SNMP and SSH packets constitute the overhead generated by the system.

V. Experimental Results

The present section discusses the experimental results performed by the system described in Section III-B when tested under the test environment described in Section IV-A. The goals are to evaluate the benefits of the achievable energy savings as opposed to QoS degradation (e.g., in reliability), and to evaluate the overhead added when operating the system. System scalability is addressed in terms of network overhead generated by the solution.Four experiments composed by different policies are evaluated and compared, showing how conflicting requirements and objectives coalesce in a decision that may compromise QoS level and energy savings.

The *first experiment (i)* has a constraint-free policy and makes use of minimal hops paths. Experiment (i) is referred to as the baseline experiment. The *second experiment (ii)* is composed by a policy that prioritizes only energy saving requirements and thus accepts QoS degradation. Within experiment (ii) saving energy is expected to be maximum. The policy of the *third experiment (iii)* ensures network performance constraints more than prioritizing energy savings. Such a policy does not accept packet loss until network reaches capacity limits. The *fourth experiment (iv)* has a policy that ensures network reliability constraints more than prioritizing energy savings. All the experiments are composed by the same network topology, traffic profile, and power profile and reliability parameters of the devices.

A. Network Topology

Figure 3 shows the network topology used in the evaluation. The topology comprises five routers connecting four end points. Two of the end points are video servers. The other two are clients. There are redundant paths when all the devices are on. A control plane that is parallel to the data plane connects SustNMS to the routers. The employed topology emulates a simple yet representative scenario of content distribution.

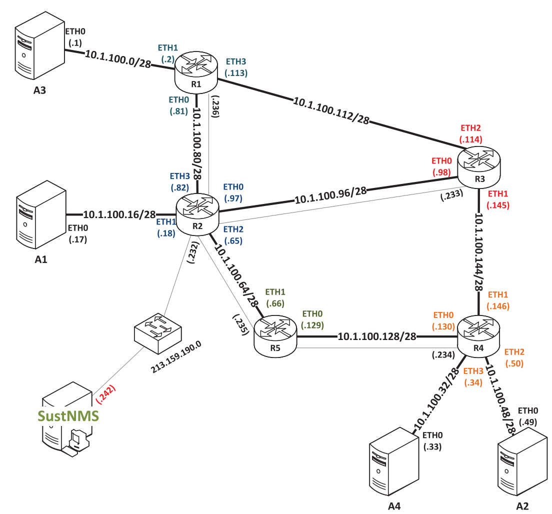

Fig. 3. Network topology equipped with the evaluation test environment described in Section IV-A

B. Traffic profile

Because IP is becoming the prevalent network technology for video transport and video is increasingly becoming a significant component of IP network traffic, the experiments make use of video streaming over IP/MPLS as source of traffic. The evaluation is based on an intermittent access from video consuming users. Incoming traffic surpasses 30Mbps at two moments of peak usage. The used traffic profile consists of four video streams that start from A1 and A3. The videos are consumed by A2 and A4 (see figure 3). The video streams start asynchronously, as shown in figure 4. At most two streams flow simultaneously from the same video server.

C. Power profile

Power profiles indicate how power consumption scales with load. Two types of power profile were used. One type is based on [16] and linearly scales with load. Such a scaling is an envisioned behavior pursued in green networking [2]. Devices

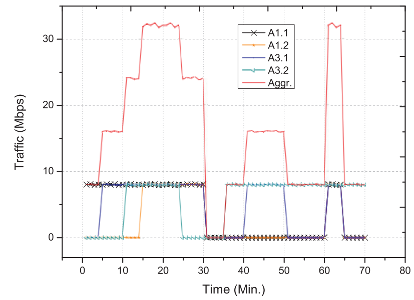

Fig. 4. Traffic profile

of such a type thus represent future equipment. The other type of power profile does not show significant variation as load increases, thus keeping an almost constant consumption. Such a type of profile is representative of profiles found in the literature, such as the one in [17]. Devices that have such a power profile represent legacy equipment, common in a heterogeneous network. Because the test environment uses a software router-based approach, the system calculates the power consumption proportionally to the maximum capacity of a corresponding physical device. The maximum capacity of the software routers were predefined as 30 Mbps for R1, R2, R3, and R5, and as 40 Mbps for R4 (see fig. 3). Because every possible stream traverse through R4 and no other router must deal simultaneously with all the possible streams, the devices were parameterized so that only R4 could hold the network maximum traffic. Maximum traffic of 32 Mbps takes place when all four streams are active. Power profiles must include a power consumption related to standby power state mode.

Four power profiles are used. Routers R1, R2, R4, and R5 scale linearly and are based on the profiles in [2], thus representing envisioned future devices. Different scaling factors are used to emulate designs for varied equipment. Router R3 has a constant power consumption (motivation from [17]). The following equations are used to indicate power consumption (Watts) as a function of usage (ratio of throughput load l to maximum capacity l_{max}):

$$P_{R1} = P_{R4} = \begin{cases} 120 & \text{if standby state} \\ 200 + l(500/l_{max}) & \text{otherwise} \end{cases} \quad (1)$$

$$P_{R2} = \begin{cases} 170 & \text{if standby state} \\ 200 + l(1000/l_{max}) & \text{otherwise} \end{cases} \quad (2)$$

$$P_{R3} = \begin{cases} 250 & \text{if standby state} \\ 1000 & \text{otherwise} \end{cases} \quad (3)$$

$$P_{R5} = \begin{cases} 220 & \text{if standby state} \\ 300 + l(5000/l_{max}) & \text{otherwise} \end{cases} \quad (4)$$

D. Energy savings

The energy savings achieved when using SustNMS were evaluated for the four aforementioned different policies. In each experiment, traffic is allocated as streaming starts, always prioritizing paths according to the constraints described in the corresponding policy.

One can enumerate four paths of the topology as follows: path 1 corresponding to (R1-R3-R4), path 2 to (R1-R2-R5-R4), path 3 to (R2-R3-R4), and path 4 to (R2-R5-R4). Paths 1 and 2 are employed in allocating traffic originated at A3. Paths 3 and 4 are employed in allocating traffic originated at A1. In the baseline experiment, devices not handling load are also kept powered on, i.e. some devices idle some times. Having devices sometimes idling allows one to perform traffic engineering and to use some device as fault protection mechanism. Figure 5 shows the power consumed by each path versus throughput, when all the network traffic is allocated to a single path.

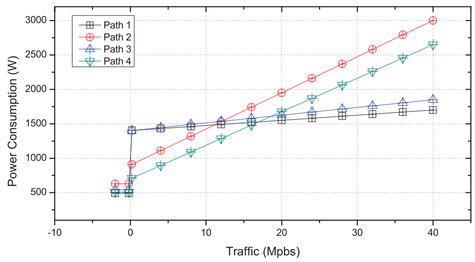

Fig. 5. Power consumption per path when all the network traffic is allocated to a single path

In fig. 5, the power consumption per path shows that the optimal allocation for energy efficiency depends on the traffic. The figure shows that, up to $\approx$ 10 Mbps throughput, the conjoint power consumption of routers in path 1 is greater than the conjoint power consumption of routers in path 2. Beyond $\approx$ 10 Mbps throughput, path 1 becomes more efficient than path 2. The following question is how much energy is saved when the optimal allocation for energy efficiency is prioritized. Such a question is addressed by using SustNMS to measure indexes, verify constraints, and thus select the optimal allocation.

In the four experiments described, SustNMS is used to allocate the traffic depicted in figure 4. In doing so, the proposed system uses Algorithm 1 for determining the optimal traffic allocation and the devices that must be put in sleep mode. The policy described in the experiments drives the decision of the algorithm. The network behavior when applying Algorithm 1 to the topology is visible in figures 6 and 7.

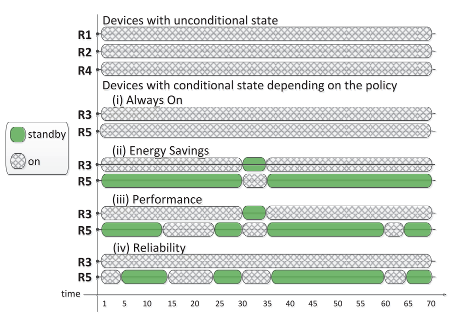

Fig. 6. Topology changes applied by the SustNMS. Probe rate of 1 min.

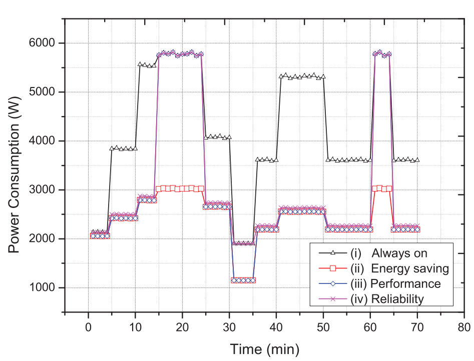

Fig. 7. Power consumption of all the described experiments

One observes in figure 6 that in every experiment the routers R1, R2 and R4 are always on, i.e. such routers never sleep. This is so because making any of them sleep implies network disconnection, which must not occur. On the other hand, routers R3 and R5 are redundant. Depending on the current traffic, one of them may be chosen for sleeping, but never both. The figure shows that, in each experiment, setting a router in or off sleep mode has distinct behavior. Such a fact exposes that, depending on which requirements are enforced, changes in topology may occur more often.

Figure 7 compares the power consumption of the non-optimized experiment (i) with the power consumption of the other three experiments. The figure shows that significant energy savings are achievable when more efficient allocation is enforced, as in experiment (ii). The allocation enforced in experiment (ii) leads to 43% of energy savings. However, the benefits come with the cost of additional control messages and an inherent network QoS degradation. Experiment (iii), which ensures performance constraints, allocates traffic in a way that leads to 30% of energy savings. Experiment (iv), which ensures constraints related to network reliability, leads the energy savings to 27%. The results show the trade-off between only prioritizing energy saving constraints and also considering QoS constraints. In all cases the amount of energy savings has a good result.

Figure 8 shows the aggregate power consumption of experiment (iii) versus probe rates. The baseline bars of the figure represent the power consumption of the baseline experiment. The figure exposes that the more frequent the probe is, the higher the energy savings are.

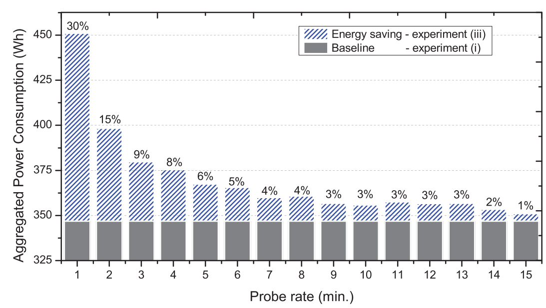

Fig. 8. Aggregated power consumption versus inter-probe interval, with energy savings highlighted. Experiment (iii).

Following, considerations on reliability and on the overhead generated by the system. Next, how such an overhead is correlated to energy savings and to the accuracy when determining the optimal traffic allocation.

E. System operation overhead

The overhead added when operating the system is related to the additional measuring and controlling packets required for, respectively, collecting data and enforcing decisions. Measuring messages are issued according to the probe rate. The relation between probe rate and how frequent the traffic varies determines the accuracy of the collected data and the effectiveness of energy gains, because the optimal paths are calculated using the current collected traffic information. For performing data collection, the system requires a total of 2 SNMP packets per probe. Such packets are request and response packets. For making an ingress router use the desired tunnel, the system requires 1 packet. For defining in which power state some device should be, the system requires 1 packet. Such a packet is a controlling packet that carries CLI commands. The presence of the above-mentioned types of packets were captured and ascertained by using proper filters with the Tshark sniffer.

Figure 9 compares the resulting overhead when operating SustNMS in each experiment, at different probe rates. The resulting overhead is related to the topology transitions the network undergoes. When going from a situation where only R5 is in standby to another situation where only R3 is in standby, two CLI commands are sent. One command is for activating R5. The other is for making R3 sleep. Probe rate of 1 minute, despite of imposing more overhead, makes the network more green, as shown in figure 10.

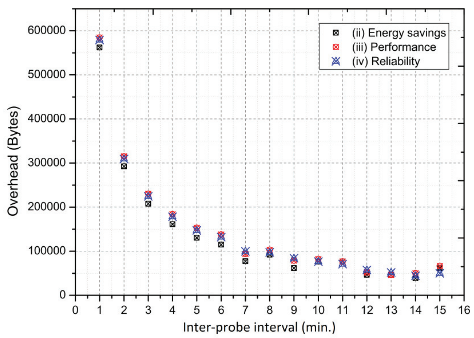

Fig. 9. Overhead due to control and monitoring packets during 70 minutes, for experiments (i), (ii), and (iii), allotted by inter-probe interval.

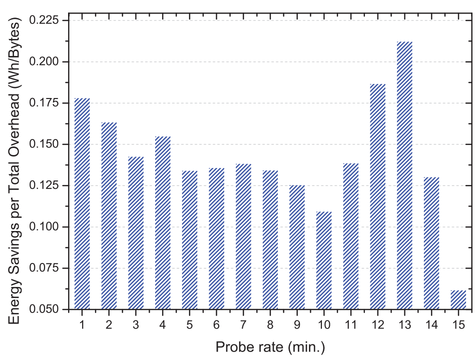

Fig. 10. Overhead divided by power consumption reduction, allotted by inter-probe interval. The data were collected for experiment (ii).

Operation overhead is lower when a lower probe rate is used. However, a low probe rate has the side effect of lowering system accuracy when determining traffic allocation. Figure 10 shows the ratio of the amount of energy savings achieved by SustNMS to the overhead when using different probe rates, for experiment (ii). The figure shows a linear tendency from probe rate of 1 minute to probe rate of 10 minutes. Probing accuracy relies on how the last measured load relates to the mean load of the subsequent inter-probe interval. For 12 and 13 min. rates, the probe instants of the experiment were not as much representative of the corresponding inter-probe time as for other rates. Thus, traffic remained underestimated for long periods, leading to non-desired losses and to higher energy savings per overhead. Fig. 10 depicts the saved energy, not the actual quality of it.

VI. Discussion

Experimental evaluation showed that energy savings of 43% can be achieved when one prioritizes energy efficiency over an always on and minimal hops allocation. Some degradation in network performance and reliability is accurately observed through an integrated real-time evaluation. However, further experiments showed that a controlled trade-off yields the practical exploitation of potential energy savings when QoS constrains are defined. Experiments showed that 30% of energy savings can be achieved when no performance degradation is allowed. Conversely, 27% of energy savings are achieved when a seven-nines reliability constraint is imposed. Such savings are conditioned to a 1 minute polling frequency and are additional to the inherent savings achieved by the energy-optimized path calculation algorithm. The overhead induced by the management system, in terms of polling for traffic counters and centralized control signaling for switching between pre-configured paths, was measured. Experiments correlating exploited energy savings to polling frequency indicates how the optimization cost relates to additional overhead.

Building a testbed for evaluating energy-efficient management algorithms is challenging. Standard Linux-based software routers lack several important features that emulate realistic network behavior. Such features are the configuration of power profiles for the software routers, emulation of the standby state, and the support of traffic steering. The testbed-based approach developed extends existing solution. On top of a Linux-kernel, it integrates standby emulation, MPLS support and policy-driven allocation decisions. The approach was successfully validated through the experiments that evaluated SustNMS, indicating the feasibility of the method in evaluating energy efficiency-oriented management systems. Possible differences between the functionalities of a software router and of a hardware router, and the usage of power profiles instead of using real time measures from routers are possible limitations of the test environment, for the accuracy of the emulation depends on the used parameters.

VII. Conclusion

We presented results from evaluating a policy-based energy efficiency network management system in a network environment emulated with software routers. The management system is centralized and reacts to changes in the network load by steering the traffic towards the most efficient communication path in accordance to operator-controlled policies. We found out that the energy savings in the presence of dynamic traffic are dependent on how often the network state is observed by the management system.

As future work, we envision evaluating SustNMS with other network topologies, in order to ascertain scalability constraints. We also envision extending the implementation by attaching to it a policy framework that enables automated policy refinement and dynamic varying policies. The policy refinement should go from the business level to the network, devices and instance levels. Dynamicity refers to changing the value of existing parameters or including new ones to the policy. Such a framework contributes to reducing the complexity in managing the network, making it simpler from a management point of view.

REFERENCES

[1] GeSI, "Smart 2020: Enabling the low carbon economy in the information age," The Climate Group on behalf of the Global eSustainability Initiative (GeSI), Tech. Rep., Jun. 2008.

[2] R. Bolla, F. Davoli, R. Bruschi, K. Christensen, F. Cucchietti, and S. Singh, "The potential impact of green technologies in next-generation wireline networks: Is there room for energy saving optimization?" *Comm. Magazine, IEEE*, vol. 49, no. 8, pp. 80 –86, august 2011.

[3] M. Zhang, C. Yi, B. Liu, and B. Zhang, "Greente: Power-aware traffic engineering," in *Proceedings of the The 18th IEEE International Conference on Network Protocols*, ser. ICNP '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 21–30.

[4] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures," *Communications Surveys Tutorials, IEEE*, vol. 13, no. 2, pp. 223 –244, quarter 2011.

[5] C. H. A. Costa, M. C. Amaral, G. Janurio, T. C. M. B. Carvalho, and C. Meirosu, "SustNMS: towards service oriented Policy-Based network management for Energy-Efficiency," in *Second IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT 2012) (SustainIT 2012)*, Pisa, Italy, Oct. 2012.

[6] M. C. Amaral, C. H. A. Costa, T. C. M. B. Carvalho, and C. Meirosu, "REASoN - REliability and/or availability evaluation for sustainable networking," in *4th Int. Work. on Reliable Networks Design and Modeling (RNDM'12)*, St. Petersburg, Russia, Oct. 2012.

[7] M. Chandramouli, B. Schoening, J. Quittek, T. Dietz, and B. Claise, "Power and energy monitoring mib," Network Working Group, IETF, Tech. Rep., May 2011.

[8] F. Maciá-Pérez, D. Marcos-Jorquera, and V. Gilart-Iglesias, "Energy management system as an embedded service: Saving energy consumption of ICT," in *Proceedings of the 22nd International Conference on Architecture of Computing Systems*, ser. ARCS '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 195–206.

[9] S. Chaudhari, S. Nottath, M. Subramanian, and H. Murthy, "Green-IT: An approach to energy savings using energy aware network management system," in *Communications (NCC), 2012 National Conference on*, feb. 2012, pp. 1 –5.

[10] H. Takeshita, Y. Oikawa, H. Yonezu, D. Ishii, S. Okamoto, and N. Yamanaka, "Demonstration of the self organized dynamic link power management by MiDORi; energy optimal network topology design engine," in *OFC/NFOEC*, march 2011, pp. 1 –3.

[11] J. Chabarek and P. Barford, "Power-awareness extensions for network testbeds," in *Communications Workshops (ICC), 2011 IEEE International Conference on*, june 2011, pp. 1 –6.

[12] J. Kempf, S. Whyte, J. Ellithorpe, P. Kazemian, M. Haitjema, N. Beheshti, S. Stuart, and H. Green, "OpenFlow MPLS and the open source label switched router," in *Proceedings of the 23rd International Teletraffic Congress*, ser. ITC '11. ITCP, 2011, pp. 8–14.

[13] W. Dongmei and L. Guangzhi, "Efficient distributed bandwidth management for mpls fast reroute," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 2, pp. 486 –495, april 2008.

[14] S. Gerdes and O. Bergmann, "Building a test environment for emulating link characteristics of disruptive networks," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, april 2010, pp. 1 –5.

[15] "Mpls for linux project." [Online]. Available: http://mpls-linux.sourceforge.net/

[16] S. Antonakopoulos, S. Fortune, and L. Zhang, "Power-aware routing with rate-adaptive network elements," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, dec. 2010, pp. 1428 –1432.

[17] A. Adelin, P. Owezarski, and T. Gayraud, "On the impact of monitoring router energy consumption for greening the internet," in *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, oct. 2010, pp. 298 –304.