Orchestration of Energy Efficiency Capabilities in Networks

Ana C. Riekstin, Guilherme C. Januário, Bruno B. Rodrigues, Viviane T. Nascimento, Tereza C. M. B. Carvalho[1], Catalin Meirosu[2]

Abstract

The energy consumption of Information and Communication Technologies is increasing and, consequently, the greenhouse gases emissions. Different approaches to cope with such challenge have been proposed, such as reducing the energy spent by a device component, or performing green traffic engineering in order to maximize the time some devices can remain in a reduced power mode. However, such energy management capabilities were designed to operate autonomously and independently from each other. When more than one is present in a node or the whole network, there is a significant potential for conflicts among them. On the other hand, there is also a potential to increase savings if more than one capability is combined properly. Moreover, if this combination takes business directives into consideration, the benefits can be more significant. In this scenario, this work proposes a novel method to orchestrate energy efficiency capabilities in an automated way. The method relies on Policy-Based Network Management and policy refinement to bring business policies down to the network operation. The method was evaluated in a Software Defined Network environment and, besides ensuring a conflict-free operation, it achieved more savings than the capabilities operating individually. In one of the experiments described in this work, a link rate adaption capability was applied after a green traffic engineering and sleeping capability. The amount of additional savings achieved was 5%, with the possibility to achieve up to 21%

[1]University of São Paulo, São Paulo, Brazil
[2]Ericsson Research, Stockholm, Sweden

of additional savings, depending on the network workload and equipment power profiles. Besides, the method can choose which combination of capabilities is the best for a given scenario, thus turning on the one that has the best results considering savings and quality of service.

Keywords: Networks; Network Management; Sustainability; Energy Efficiency; Policy Refinement

1. Introduction

It is a known fact that the energy demanded by Information and Communication Technologies (ICT) is increasing worldwide annually. It is so that studies [1] [2] [3] indicate that in the next five years the consumption of the sector will reach 1,100 TWh. In the U.S. alone, ICT facilities are responsible for 120 TWh of energy annually, corresponding to 3% of all U.S. demand. The country is the second in energy consumption, demanding nearly the same amount of energy as China and four times that of Japan, ranked in the third place [5]. Attached to the energy demand is also the problem of greenhouse gases (GHG) emissions, and the users' increasing concern with the companies' responsibilities [4]. Worldwide, ICT is responsible for 2% of the carbon emissions [6], a figure that is predicted to grow to 2.3% by 2020.

Within the ICT sector, datacenters, embodied by servers, networking, and cooling, is the fastest growing source of energy consumption. Its demand grew 7% in 2013, when compared to the previous year, nearing 350 TWh [5]. The prediction is that it will have grown 81% by 2020. However, how much of that amount corresponds to the network is not actually consensus: 4% in [7], 12% in [8], one third in [9], 9% in [10], 23% in [11], 22% in 2011, projected to 24% in 2020 in [5]. Even if not consensus, considering the current energy efficiency efforts on the other parts of the datacenters, the share of networking can become much higher, with the potential to raise its ratio up to about 50% [8]. The numbers can be even more significant for telecom operators, for whom the energy costs are among the most relevant [1]. Considering the associated GHG emissions, Verizon reported that the electricity to run its networks surpassed 92% of their total carbon emissions in 2013 [12].

To mitigate such environmental costs, more energy efficient networking devices and techniques have been devised. However, such capabilities not always work together properly, thus lacking a specialized management layer to allow a better exploitation of their combined benefits. In this article, we show a novel method to orchestrate energy efficient capabilities of the network. Our method relies on Policy-Based Network Management (PBNM) and policy refinement to shorten the distance between management level and device-level operation, therefore making energy savings in the network more straightforward from the management point of view.

Examples of efforts in energy-efficient networking that tackle a single device and its immediate neighbors are the Adaptive Link Rate (ALR) [13] and the techniques of Synchronized Coalescing, and Adaptive Coalescing [14]. At the network level, there are the energy-aware routing mechanism proposed in [15], Green Traffic Engineering (GreenTE) [16], Sustainability-Oriented Network Management System (SustNMS) [17], and ElasticTree [18], which targets Software Defined Networks (SDNs). Though straightforward the operation of a single functionality might be in a homogeneous network, the necessary orchestration can be costly in a more heterogeneous environment, with functionalities varying among nodes or with nodes with more than one functionality enabled. This can be even more challenging if we consider the alignment with business policies.

The expression of business-level policies and its subsequent translation to device-level actions and configuration increases the automation level of the network management, turning it less error prone and complex. This can be achieved through Policy-Based Network Management (PBNM). With the aid of PBNM, network managers can provide users with green Service-Level Agreements (SLAs), thus offering green services and products. These would be ultimately implemented as sustainability-oriented policies that manage the energy efficiency functionalities of the network. This way, a network operator can foster a reduction in GHG emissions and energy expenses. A PBNM scheme can be comprised of abstraction levels other than the two focused on business expressions and actions and configurations of devices. The translation between such levels is called Policy Refinement and, although this has been studied before, the lack of a standard [19] just accrues to the inherent difficulty of refining novel high-level sustainability-oriented policies, either in legacy networks or in more modern SDNs. For further insights in such challenges and related requirements, we refer the reader to [20].

In this article, we detail a method devised to orchestrate energy efficiency functionalities in a network, comprising the refinement of sustainability-oriented policies. We also show how a proof-of-concept of the proposed method was prototyped and then validated. Having as start point the business level and sustainability-oriented information models, randomically generated workloads, the network topology, the power profile of the devices, and the knowledge of the deployed energy efficient functionalities, the method generates an interpolated decision tree against which decisions are made. Decisions are such as which routing or local energy-efficient technique to apply, or a combination thereof, in which period of the day, under which network conditions.

To the best of the author's knowledge, this is the first method to refine sustainability-oriented policies from business level down to network level and orchestrate energy efficiency functionalities, thus enabling a more energy efficient and automated network infrastructure. In detailing our method, for the first time we show a sequence diagram and elements of a structured language to be used with the implementation architecture first presented in [21]. The intrinsics and decision steps of the method are also discussed and exemplified for a given network and power profiles.

The remainder of this article is organized as follows. Section 2 provides a common ground for further discussion and describes related work. Section 3 details the open issues considering the existing solutions. Section 4 details our method and has the bulk of the theoretical and general content, whereas in Section 5 we show how we implemented our proof-of-content system to save energy in the network. The obtained results are reported in Section 6. Discussions and concluding remarks are drawn respectively in Sections 7 and 8.

2. Background

Several functionalities and protocols have been proposed to cope with energy efficiency in networks. Such capabilities can be separated by the specific scope: component of a device, a complete device, or the whole network [22]. Adaptive Link Rate (ALR) [13] and Advanced Configuration and Power Interface (ACPI) [23] are examples of capabilities applied on specific chip-level components of the network devices. Synchronized Coalescing (SC) [14] and IEEE 802.3az [24] are implemented at the node level on the network elements, and there are capabilities that are employed at the network level, such as Green Traffic Engineering (GreenTE) [16] or SustNMS, a sustainability-oriented network management system, which operates based on policies focusing on the trade-off between energy efficiency and Quality of Service (QoS) [17].

In order to bring business directives to these energy efficiency capabilities, policy-based network management (PBNM) concepts can be used. A policy is a set of rules that chooses a response to a specific condition in order to enforce behavioral or functional actions [25]. A sustainability-oriented policy is a policy that manages energy efficiency features in the network [20]. Policies can have different levels of abstraction, starting at the business level, and going down to device or instance levels [26]. The translation between the different levels of abstraction is called Policy Refinement.

There are two main categories of complete policy refinement methods: transformation rules and goal-based approaches [20]. The first category comprises methods that use pre-defined rules to translate high-level policies down to machine-readable policies. They can also include mechanisms to support such task, as resources discovery and policy analysis (coverage and conflicts detection and resolution). Within this category, we can cite Verma [27]. The author proposes to use tables do relate users, applications, and devices to classes of service. The method performs table lookup to build the relationships during the refinement, thus depending on the correctness of the table contents. This drawback is compensated by easiness of analyzing contradictions and coverage of such a rule-based notation. Verma's is the most automated solution for policy refinement, although domain specific (for QoS policies). The second category comprises the methods that use goals in the translation process. Bandara et al. [28] propose a not automated method based on Event Calculus. The first step is to refine the goals to operationalized goals, followed by the mapping of specific operations that can be implemented. The environment is modeled using UML. Rubio-Loyola [29] automates parts of this method by using a model checker to determine the actions to be performed, translated after to Ponder2 [30] instructions. Craven et al. [19] is a newer goal based approach in which the domain is modeled using UML, based on four steps: policy decomposition, operationalization, re-refinement and deployment of policies using Ponder2.

3. Problem Statement

Energy management capabilities were designed to operate autonomously and independently from each other. When more than one of them are present in a network node or the whole network, there is a great potential for conflicts among these capabilities. Such conflicts could reduce or negate energy savings, or even lead to undesired behavior, such as repeatedly turning on a node or putting it to sleep.

Both categories of refinement methods (translation rules or goal-based approaches) focus on translating the policies from a high to a lower level of abstraction. They support policy analysis [31], resources discovery [27] and dynamicity of time or, with some adaption, scenario changing (e.g., when a node is migrated to another location), as the adaptation by changing policy parameters according to the situation [32]. Such methods could be extended to comprise energy efficiency capabilities translation. However, they do not orchestrate energy efficiency capabilities. That is, they are not able to combine more than one capability to save more energy, nor to do this in a conflict-free operation. Note that conflicts among policies differ from conflicts among energy efficiency capabilities. For instance, the first will occur between two policies, one stating to save energy, while other asks for more performance; the second deals with specific conflicts among green energy efficiency functionalities (e.g: a capability that puts a device to sleep while other tries to wake it up).

Orchestrating energy efficiency capabilities, besides operating the network in a conflict-free manner, allows to combine more than one capability at the same time in the nodes. Such coordination could lead to bigger savings than enabling just one functionality. Besides, such method should be able to choose which capability (-ties) is (are) the best for a given scenario, thus turning on the one that has the best results considering savings and quality of service.

There is no method available that can orchestrate or coordinate energy efficiency capabilities, nor to consider business directives in such operation in an automated way. Such orchestration demands additional and specific information on how the energy efficiency capabilities work, their scope and type (sleeping or rating), besides being able to calculate energy savings and losses for each capability(-ties) in order to define the best combination of capabilities for a given scenario.

4. Proposed Method

The Sustainability-Oriented System (SOS) orchestrates different energy management capabilities of the network infrastructure taking business directives into account. The method sequence diagram is presented on Figure 1. The first step is the definition of the business policies by the network operator, based on an object-oriented information model that comprises all objects and parts of a policy [33]. There are three information models for the different levels of abstraction a policy can have: Business/System Levels, Network Level, and Device Level.

The inclusion of the business policies is done through the SOS Interface, and the definition comprises the information about the period of the day that the

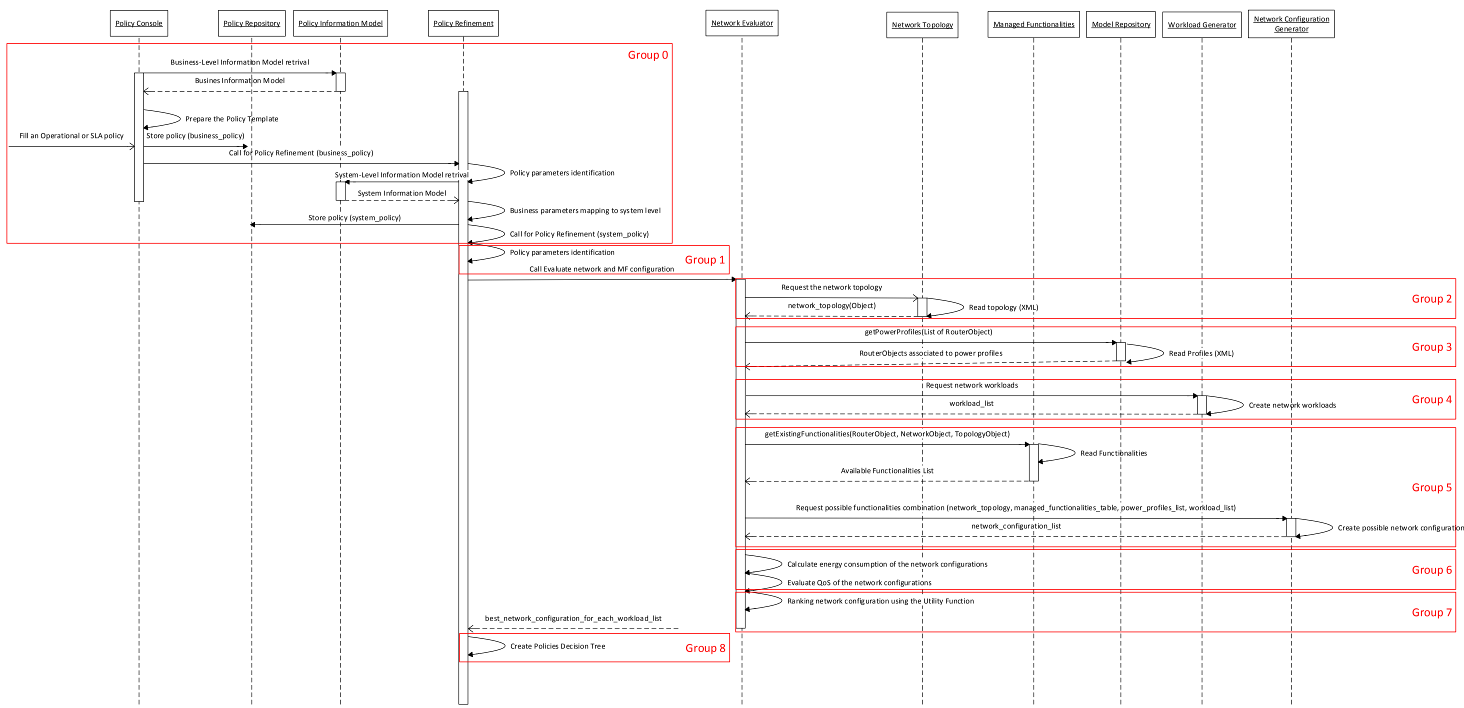

Figure 1: SOS Sequence Diagram

capabilities can be applied, the bandwidth utilization condition for the use of the capabilities and if it is desirable by the operator to employ or not the energy efficiency capabilities on the network infrastructure. These inputs on the SOS Interface are named Environment Condition, Time Condition and Action, all of them represented in the highest level of the information models.

The translation of the business policies uses a Table Lookup approach that translate the parameters based on information contained on previously defined tables (the representation of the information models). The tables for the method are in XML format due to the easiness of implementation and translation of the information contained on the tables. The module contains one XML table for each business input existing on the interface: Environment Condition, Time Condition, and Action. Each one of the business policies received from the previous level is compared with the content of the respective table. The identified translated information is selected from the table, and this translated data is saved on a repository for future use. These steps are represented in the Group 0 in Figure 1.

Examples of refined policies in the network level are:

$$
\begin{aligned}
&[' NetworkCapacity == 1000 \ and \ load \ < \ 200.0',\\
&'time >= 22 \ and \ time < 6',\\
&'apply \ any \ possible \ capability(-ties)']
\end{aligned}
$$

$$
\begin{aligned}
&[' NetworkCapacity == 1000 \ and \ load \ < \ 800.0',\\
&'time >= 6 \ and \ time < 226',\\
&'apply \ only \ link \ rating \ capabilities']
\end{aligned}
$$

The translated policies allow the identification of parameters necessary for the network to be configured (Group 1 in Figure 1). The next action is the determination of which capability or combination of capabilities will be employed on the network structure. The definition is performed based on information about the topology of the network (Group 2 in Figure 1), the equipment power

Table 1: Energy Efficiency Capabilities Classification

	Capability		Scope		Topology		
	Rating	Sleeping	Device	Network	SoHo/LAN	WAN	Fat Tree
ALR	√		√		√	√	√
SC		√	√		√	√	√
SustNMS		√		√	√	√	
ElasticTree		√		√			√

profiles (Group 3 in Figure 1), the bandwidth utilization values (Group 4 in Figure 1) and the available functionalities (Group 5 in Figure 1). Power Profiles are equations representing the equipment power behavior (in Watts) for a given workload. The main objective in the current step is to build a preliminary tree (Preliminary Tree A, or PTA) in which the leafs correspond to a set of random bandwidth utilization values, as illustrated on Figure 2. For each bandwidth utilization, the method tests the allowed capabilities combination as a way to train our method to be able to give the best answer for a given bandwidth utilization value in the target topology.

The differences of the capabilities existing on the network and their specifications must be treated to avoid possible conflicts on their use. The next module requests the possible capabilities combination. Table 1 shows the classification of some example capabilities.

Associated with this table, there are some policies to determine the possible combinations:

- Two network level functionalities performing similar actions cannot be used at the same time;
- Two or more functionalities to put devices to sleep cannot be applied at the same device, at the same time;
- Two functionalities to do link rate cannot be applied at the same device, at the same time.

These policies, combined with the capabilities information in Table 1, are defined in the system as an auxiliary table, the "Allowed Combinations Table". This table is consulted in the step "Request possible functionalities combination" (Group 5 in Figure 1). The possible capabilities or combinations, considering the ones exemplified are:

- ALR in the interfaces of all nodes;
- Synchronized Coalescing in all nodes;
- Combination of ALR in the interfaces and Synchronized Coalescing in the nodes;
- SustNMS in the network;
- ElasticTree in the network;
- SustNMS in the network, plus ALR in the interfaces that remain powered on;
- ElasticTree in the network, plus ALR in the interfaces that remain powered on.

For each leaf of the preliminary tree PTA, the method put together the information about the bandwidth utilization for that leaf and the equipment power profile to get the amount of electrical energy expected to be spent in a baseline operation mode. The same module, which we call Analytical Solver (Group 6 in Figure 1), calculates the expected energy consumption and packet loss for each bandwidth utilization applying the capabilities allowed combination.

In more detail, the Analytical Solver needs the following inputs: *(i1)* Parameters necessary for the network to be configured (Group 1 in Figure 1); *(i2)* Topology of the network (Group 2 in Figure 1); *(i3)* Equipment power profiles (Group 3 in Figure 1); *(i4)* Randomly generated bandwidth utilization values (Group 4 in Figure 1), organized as a preliminary tree (PTA) with the leafs corresponding to the different bandwidth utilization scenarios, as exemplified in Figure 2. The amount of leaves in the tree must be defined before generating the random bandwidth values. In our method, the number of leaves is defined

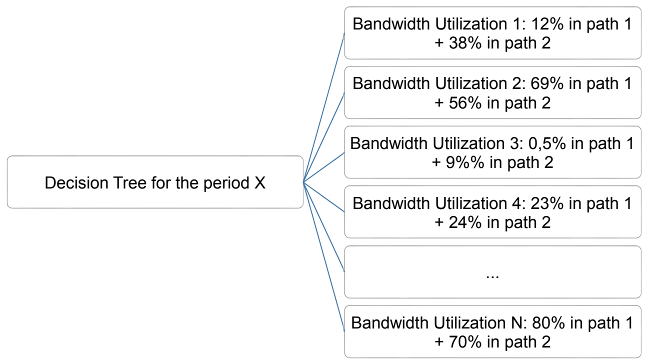

Figure 2: Preliminary Tree A (PTA) Example, supposing two paths available

empirically, by partitioning the interval between zero and the maximum value of available bandwidth in an arbitrary number of sub-intervals. A person skilled in the art may use other strategies, noting that since the method uses interpolation in the later stage, increasing the number of leaves gives more accuracy. That is, the more leaves, the more likely is that the method will be applying the best capability (calculated by a utility function, as described below) for the instantaneously measured bandwidth value.

After deciding how many bandwidth values the tree will have, the method defines the exact sub-intervals by executing a random numbers generator and considering the refined policies. For instance, if the policy says to apply capabilities only if the bandwidth is smaller than 50% of the maximum workload of the network, the method will generate only random numbers smaller than $maximum\ load * 50\%$. The last input, *(i5)* is the available energy efficiency capabilities along with the possible combinations information (Group 5 in Figure 1).

The Analytical Solver processing part will perform the steps described in the Algorithm 1 (Group 6 in Figure 1). The output is the Preliminary Tree B (PTB) composed by leafs with bandwidth utilization values and a set of capabilities along with the expected savings and packet losses, as exemplified in Figure 3.

Based on the estimated values of savings and packet losses for the topology

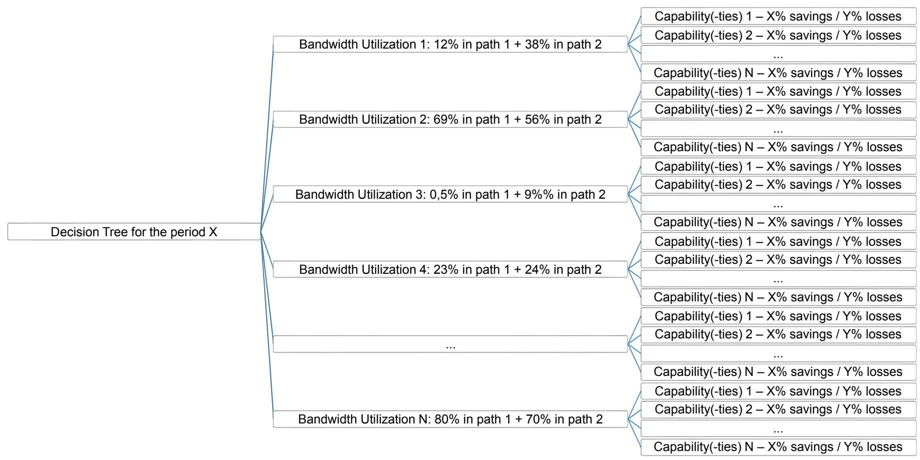

Figure 3: Preliminary Tree B (PTB) Example with expected savings and losses for each bandwidth utilization and associated capability(-ties)

and the equipment power profile calculated for each leaf of PTB by the Analytical Solver, the Utility Function (UF) determines the grade for each energy efficiency capability(-ties) combination using the following formula (Group 7 in Figure 1):

$$UF = pl * \frac{1}{\frac{\sum_{k=0}^{n} EnergyAfterSavings_{RouterK}}{\sum_{k=0}^{n} EnergyBaseline_{RouterK}}} \tag{1}$$

Algorithm 1 Building the Preliminary Tree B (PTB)

Require: $Inputs\ i1, i2, i3, i4, i5$

  for all bandwidth utilization value (leaf) in the preliminary tree PTA **do**

    for all allowed capabilities combination **do**

      Estimate energy savings based on power profiles for each of the available energy efficiency capabilities, applied to each generated bandwidth utilization value.

      Estimate packet loss for each bandwidth utilization value and associated capability(-ties) under evaluation.

    end for

  end for

Packet Loss range	"pl"
$0\% < x < 0.1\%$	1
$0.1\% < x < 1\%$	0.9
$1\% < x < 30\%$	0.8
$30\% < x < 40\%$	0.7
$40\% < x < 50\%$	0.6
$x > 50\%$	0

Table 2: Packet Loss Categories

This Utility Function combines energy savings and QoS, assuring a minimum level of service while trying to save energy, for an n-router topology. The term *pl* refers to packet loss. The amount of losses is grouped in six categories as described in Table 2. The more packets lost, the less the value of the Utility Function, and, if more than 0,5% of the packets are lost, the value is zero, ensuring the QoS level and connectivity. The categories can be changed according to the network services provided. The higher the grade a capability achieves with the Utility Function, the better is the combination of this functionality (-ties) for the given bandwidth utilization and topology.

Considering the Utility Function result, the method will build the Final Decision Tree (FDT), in which each leaf is composed by a bandwidth utilization value and an associated capability(-ties), the one(s) which achieved the highest grade in the Utility Function, as illustrated in Figure 4. For each period of the day, one different Final Decision Tree is expected.

The Final Decisions Trees are received by the last module of the method, which will train the network to react differently considering the bandwidth utilization (Group 8 in Figure 1). The required inputs for this step are: *(i6)* Final Decision Tree FDT composed by leafs with bandwidth utilization values and

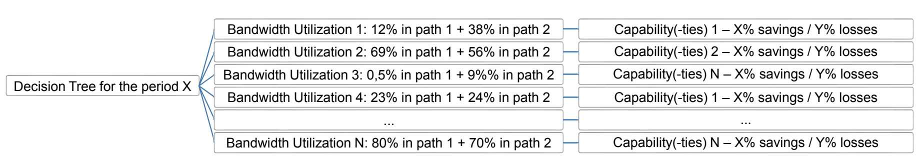

Figure 4: Final Decision Tree (FDT) Example with the best capability(-ties) for each bandwidth utilization

the best capability(-ties) for the bandwidth utilization, as exemplified in Figure 3; *(i7)* the network itself equipped with a Network Management System to provide metrics such as bandwidth utilization, energy, and packet loss. After, the method builds the Final Decision Tree using an Interpolation tool (FDTI). This tree has three levels: the root, the leaves with bandwidth utilization values and, associated to each bandwidth utilization value, one leaf with the best capability (-ties) for the given bandwidth scenario. The method uses the class DecisionTreeClassifier from the Scikit tool [34] to train the FDT so that it will be able to interpolate bandwidth values, that is, predict the target values when no existing bandwidth leaf matches exactly the bandwidth measured during operation. This constitutes the Final Decision Tree with Interpolation (FDTI), exemplified in Figure 5. After this step, the method can handle all bandwidth values, even those that were not specifically calculates during the method. The Algorithm 2 describes the FDTI usage while operating the network.

To apply the capabilities in the network means issuing a set of Netconf or OpenFlow instructions that will configure the management parameters, such as link rate or state (sleep/powered on). For instance, applying ALR in the nodes

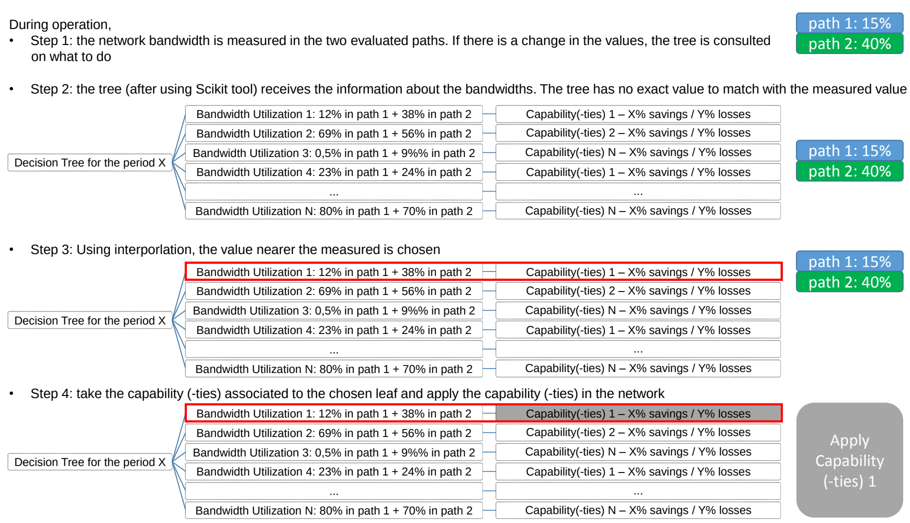

Figure 5: Example of the Final Decision Tree with Interpolation (FDTI) operation

Algorithm 2 SOS Operation

while TRUE **do**

Measure the network bandwidth utilization

if the bandwidth utilization in the network changes (Step 1 in Figure 5) **then**

Use the trained tree FDTI to determine, by interpolation, which leaf (representing an energy efficiency capability) is going to be used with the measured bandwidth utilization (Step 2 in Figure 5).

Use interpolation (or some other distance calculation method) to determine the leaf in the FDTI that is the closest match for the value of the measured bandwidth (Step 3 in Figure 5).

Take the capability (-ties) associated to this leaf (Step 4 in Figure 5).

Apply the capability (-ties) in the network (See details below).

Optional step: Measure and show savings.

Optional step: Measure and show losses.

end if

end while

means changing the link rates between pairs of nodes using Ethernet data rates (e.g. 10 Mbps) after a handshake between each pair of nodes. Apply SC followed by ALR is a two-step process. It starts by configuring the nodes to perform traffic bursts and sleep while buffering data. This configuration involves SC parameters as tOn (time the equipment will be fully operational) and DutyCyle (percentage of time the equipment must remain sleeping). Secondly, ALR will be applied during the fully operational (not sleeping) periods of SC, reducing the link rate. Apply SustNMS in the network means considering all existing paths in the topology, performing green traffic engineering to consolidate traffic in some of the paths and then putting the unused devices to sleep.

5. Implementation

Figure 6 illustrates the modules implemented, in a Software Defined Network (SDN) environment. The method was implemented in Python 2.7, using XML to represent the information when necessary. The first steps comprise the Table

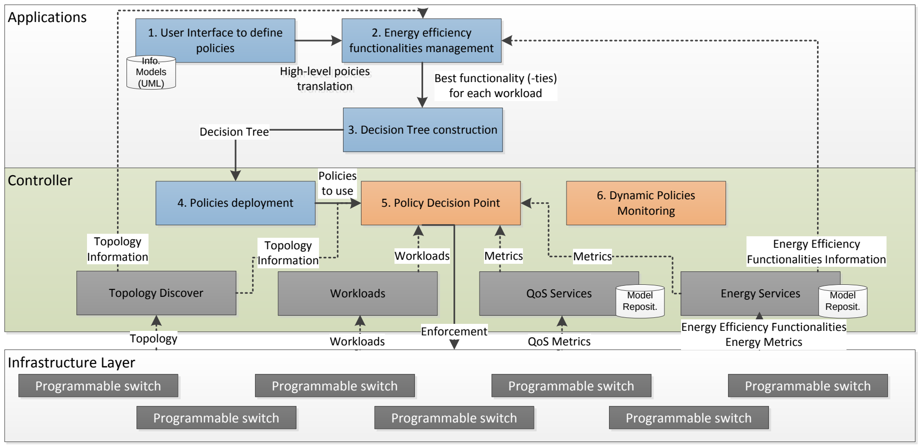

Figure 6: Implementation Modules [21]

Lookup translation, which translates high-level policies to the network level using tables to relate objects. The relationship among the objects of the policies is defined in the information models. The user defines the high-level policies in the SOS Interface. Module 1 translates the high-level policies defined by the user in the Interface using XML files defining the Environment Condition, Policy Time Condition, and Green Plan Action. The output (refined policies) is recorded in a text file, used as an input in the next module, which needs information from the high-level policies. This concludes the steps performed in the Module 1 in Figure 6.

The first steps of Module 2 are related to information gathering so that the best capability (-ties) can be selected. The network topology is represented in an XML. The devices power profiles, necessary to calculate the amount of Watts each equipment dissipates, are also represented in XML with an associated module to read and return the power profiles information from the XML. The Power Profiles used in this work were based on [35].

The next step is to generate the random bandwidth utilization values used to build the first preliminary tree (PTA). The Algorithm 3 describes the bandwidth utilization values generation. The output is another XML file, relating the topology with the generated workloads, as represented in Figure 2.

The only information still missing to the Module 2 is the possible combinations of energy efficiency capabilities, the "Allowed Combinations Table". The combinations are implemented as extra XML tables:

$functionalitiesCombinationAllCapabilities$

$functionalitiesCombinationOnlyLinkRating$

The first depicts all the combinations considering all functionalities are allowed. The other shows what can be done considering that only link rating is allowed (in case of a policy for the day, for instance, with more traffic to handle).

With all the information ready to use, the Module will use the Analytical Solvers to estimate savings and losses for each pair workload/capability (-ties). Considering the existing capabilities available, we selected three to be implemented, each one as a representative of a different scope, as described in Section 2: ALR (component scope), SC (device scope) and SustNMS (network scope). ALR puts interfaces to sleep considering the Ethernet rates: 1Gbps, 100Mbps, and 10Mbps. According to Ricca et al. [36], ALR can save up to 21% in the studied equipment. Ricciardi et al. [37] studied the capability and discovered

Algorithm 3 Random bandwidth utilization values generation

$R1 \leftarrow loadmax\ of\ the\ first\ Router$

{Distributes the load according to the first router}

for all Paths in the Topology **do**

  for all Nodes **do**

    $random \leftarrow a\ random\ number\ between\ 0\ and\ 1$

    $loadcurrent \leftarrow loadcurrent + R1 * random$

  end for

end for

{Adjusts according to the maxload of each node in the path}

$aux \leftarrow 1$

for all Nodes **do**

  $aux \leftarrow max(loadcurrent\ or\ aux)/loadmax$

  $loadcurrent \leftarrow loadcurrent/aux$

end for

that the energy spent after reducing the link rate depends on the interface native speed. The authors also state that half of the energy is due to the baseline consumption of the equipment, and that, using ALR, the savings could reach 15%. ALR is interesting to use in scenarios in which the load is bigger, since it spends much less time to wake up the interfaces (microseconds order of magnitude, while waking up a node from a sleep mode can take minutes). It is also interesting to use over other functionality, in those nodes that remain powered on, saving more, without conflicting. The ALR Analytical Solver is described in the Algorithm 4. The SC Analytical Solver is described in the Algorithm 5. The SustNMS Analytical Solver followed the algorithm described in [35].

The combination ALR/SC also has an analytical solver, as described in the Algorithm 6. The combination SustNMS/ALR was implemented in another way: the SustNMS Analytical Solver outputs the list of necessary routers to remain powered on. Over this list, ALR is applied if possible, when the workload is smaller than 10 Mbps. So the solvers remained separated. The Module 2 now has all the information necessary, as illustrated in Figure 3. The method then applies the Utility Function to choose the best capability (-ties), as described in Figure 4. This concludes the Module 2 steps.

Module 3 builds the decision tree (one or more than one, according to the number of policies defined in the user interface). The Module uses the infor-

---

Algorithm 4 ALR Analytical Solver

Require: $Inputs\ packets\ per\ second,\ power\ profiles$

  $minRate \leftarrow 10Mbps$

  for all Devices **do**

    $maxRate \leftarrow maximum\ device\ capacity$

    if Packets per second < minRate **then**

      $result \leftarrow power\ consumption\ with\ ALR\ savings\ based\ on\ power\ profile$

    else

      $result \leftarrow power\ consumption\ device\ normal\ rate$

    end if

  end for

Algorithm 5 SC Analytical Solver

Require: $Inputs\ packets\ per\ second,\ power\ profiles$

$tOn \leftarrow duration\ of\ the\ period\ with\ the\ device\ active\ in\ milliseconds$

$DutyCycle \leftarrow percentage\ of\ cycle\ time\ the\ device\ must\ remain\ active$

{tOff = (tOn / DutyCycle) - tOn}

$threshold \leftarrow number\ of\ packets\ to\ deactivate\ SC\ (adaptive\ behavior)$

for all Devices **do**

  if Packets per second < threshold **then**

    $resultOn \leftarrow power\ consumption\ device\ on$

    $resultOff \leftarrow power\ consumption\ device\ off$

    $result \leftarrow resultOn * tOn + resultOff * tOff$

    $buffer \leftarrow size\ of\ the\ buffer\ in\ number\ of\ packets$

    if packets per second > buffer **then**

      $Calculate\ packet\ losses$

    else

      $No\ losses$

    end if

  else

    $result \leftarrow power\ consumption\ device\ on$

    {Does not sleep}

  end if

end for

mation from Module 2 and calls the Scikit Tool to build the Final Decision Tree with Interpolation (FDTI), as depicted in 5, which is then deployed in the controller. From this point on, the Algorithm 2 is operating on the network.

6. Experimental Results

To test and validate the proposed method, this work used the GreenSDN testbed [38]. The network is emulated on the Mininet Virtual Machine using the POX controller to manage its actions and the Iperf tool to generate the traffic across the network. The communication between the data-plane and

Algorithm 6 SC in conjunction with ALR Analytical Solver

Require: $Inputs\ packets\ per\ second,\ power\ profiles$

$tOn \leftarrow duration\ of\ the\ period\ with\ the\ device\ active\ in\ milliseconds$

$DutyCycle \leftarrow percentage\ of\ cycle\ time\ the\ device\ must\ remain\ active$

{tOff = (tOn / DutyCycle) - tOn}

$threshold \leftarrow number\ of\ packets\ to\ deactivate\ SC\ (adaptive\ behavior)$

for all Devices **do**

  if Packets per second < threshold **then**

    $resultOn \leftarrow power\ consumption\ device\ on$

    $resultOff \leftarrow power\ consumption\ device\ off$

    $result \leftarrow resultOn * tOn + resultOff * tOff$

    $buffer \leftarrow size\ of\ the\ buffer\ in\ number\ of\ packets$

    if packets per second > buffer **then**

      $Calculate\ packet\ losses$

    else

      $No\ losses$

    end if

    $minRate \leftarrow 10Mbps$

    $maxRate \leftarrow maximum\ device\ capacity$

    if Packets per second < minRate **then**

      $result \leftarrow power\ consumption\ with\ ALR\ savings\ during\ SC\ tOn$

    else

      $result \leftarrow power\ consumption\ device\ normal\ rate$

    end if

  else

    $result \leftarrow power\ consumption\ device\ on$

    {Does not sleep}

  end if

end for

the controller is done by the OpenFlow 1.0 protocol. The POX controller is implemented using Python 2.7.

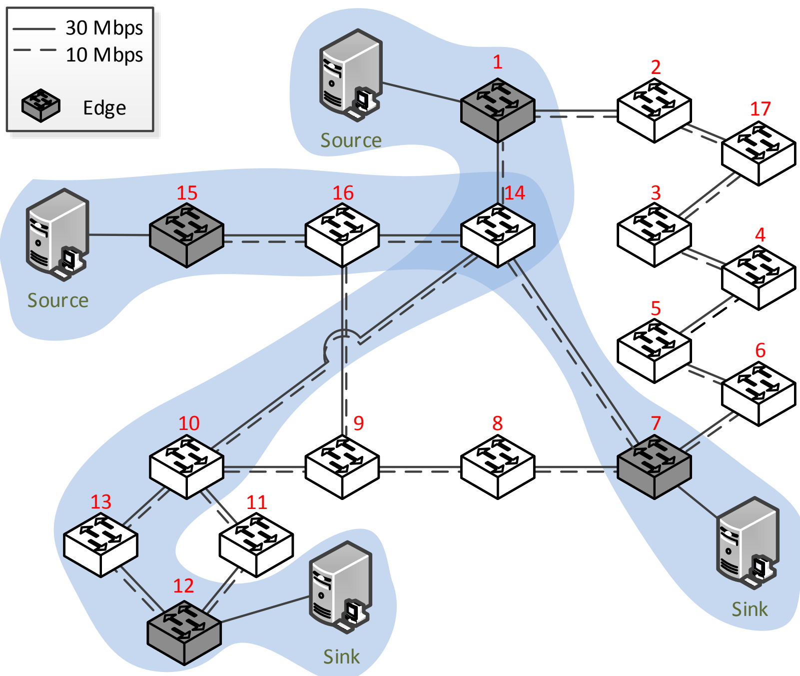

Figure 7: The topology used in the SOS method proof-of-concept

In the experiments, a load proportional power profile (PP) was used for all nodes based on [35] as described previously. There is also a power profile for sleeping periods and adapted power profiles for ALR and SC:

$$PP = 200 + \left(\frac{500}{30}\right) * workload \tag{2}$$

$$PP_{sleeping} = 120 \tag{3}$$

$$PP_{ALR} = 200 + \left(\frac{500}{30}\right) * workload - 15\% * ALR \tag{4}$$

$$PP_{SC} = PP_{sleeping} * tOff + PP * tOn \tag{5}$$

The network topology of the experiments, depicted in Figure 7, was inspired by the core part of the RNP Network (Rede Nacional de Ensino e Pesquisa, the Brazilian research and education national network).

6.1. Utility Function Validation

In order to check the Utility Function (UF) selection, we evaluated the Analytical Solvers results applied to it (ALR, SC, SustNMS Sustainability Policy, or SustNMS Performance Policy). In some cases, the maximum grade is achieved by more than one scenario that is, a pair bandwidth utilization value/capability(-ties). There is a good example of the Utility Function selection in the scenarios depicted on Table 3. This table lists all possible combinations of capabilities for one bandwidth utilization value. In this example, applying SustNMS-Sustainability saves more energy, with the final power dissipated 2663.45W, smaller than 2786.91W from SustNMS-Performance. However, SustNMS-Sustainability presents packet losses and, for this reason, loses points with the pl value 0.9 As the final result, the SustNMS-Performance Utility Function grade gets bigger, and so this option is selected.

Capability(-ties)	**With savings (W)**	**Baseline(W)**	**PL**	**UF Result**
ALR/SC	3371.95	3371.95	1	1.00
SustNMS Perf. Policy	2786.92	3371.95	1	1.21
SustNMS Sust. Policy	2663.46	3371.95	0.9	1.14

Table 3: Utility Function validation example

6.2. Orchestration Experiment of Two Capabilities Combination

To check the combinations of capabilities and respective savings and losses, we emulated an environment in which SustNMS is applied alone and in conjunction with ALR. The experiment was performed by running two flows, one from the leftmost source in Figure 7, to the sink on the rightmost side, and another from the topmost source to the bottommost sink. Figure 8 depicts the Analytical Solver results for four situations:

- The baseline with no load;
- The baseline without energy efficiency capabilities being applied with two 10 Mbps flows;

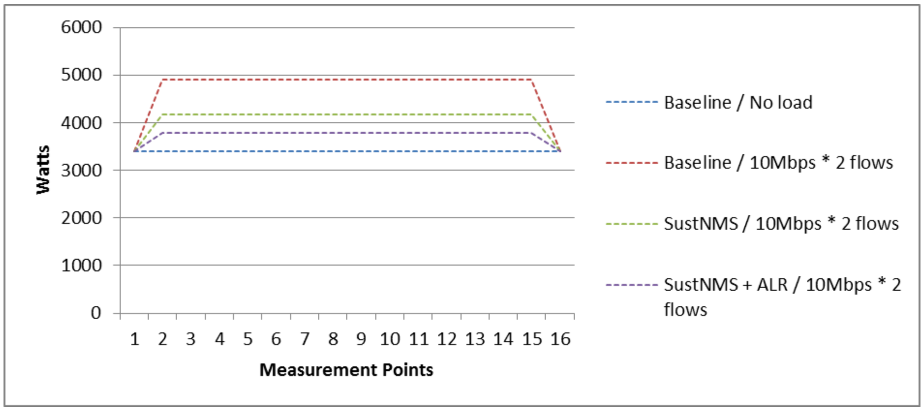

Figure 8: Analytical solver results for two 10 Mbps flows

- SustNMS being applied in the scenario with two 10 Mbps flows; and
- The orchestration scenario, in which SustNMS is applied in conjunction with ALR. This happens because the 10 Mbps workload allows ALR application, bringing more savings.

Figure 9 represents the same situations, but measured directly in the emulation environment. One can see that the results are similar. In both cases, the application of just one capability brings 15% of savings, while applying both bring 20% reduction. The amount of savings brought by energy efficiency orchestration can be as high as the maximum savings brought by the second capability. In our example, ALR was applied only after SustNMS, only in the switches that remained power on. Besides, we assumed a conservative scenario in which ALR can save 15% of energy, but this number can reach 21% [37].

This experiment considered energy proportional equipment. If the power profile is not load proportional, the savings can be more significant. Figure 10 depicts a scenario in which every switch spends 1000W regardless the load. The savings applying SustNMS is 47%. Orchestration brings 53% savings.

One important consideration is the scenario with high loads. For example, two 20 Mbps flows. Energy efficiency capabilities are motivated by the idle periods in the network. Therefore, it is expected that they will not present

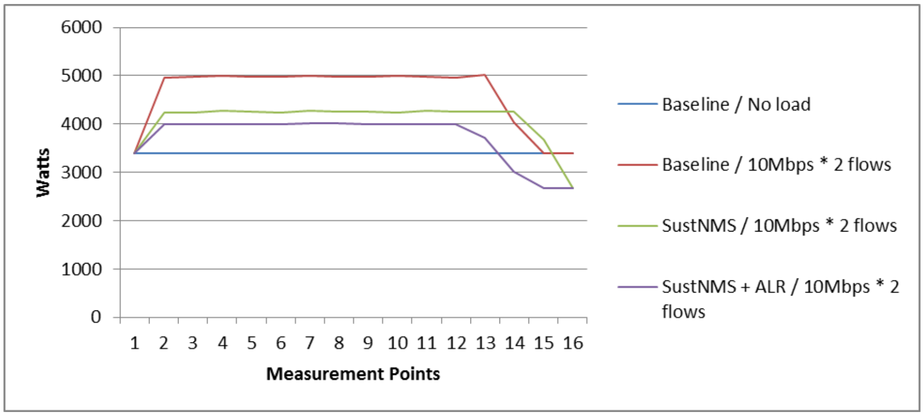

Figure 9: Emulation results for two 10 Mbps flows

significant savings in high load scenarios. Besides, ALR, for instance, would not make any difference, since it reduces link rates only according to the Ethernet rates, and, in this case, the workload is higher than the 10 Mbps reduced link rate. We used a dashed line for the SustNMS with ALR scenario to show it has the same results as applying only SustNMS, as depicted in Figure 11 The power profile used was the load proportional.

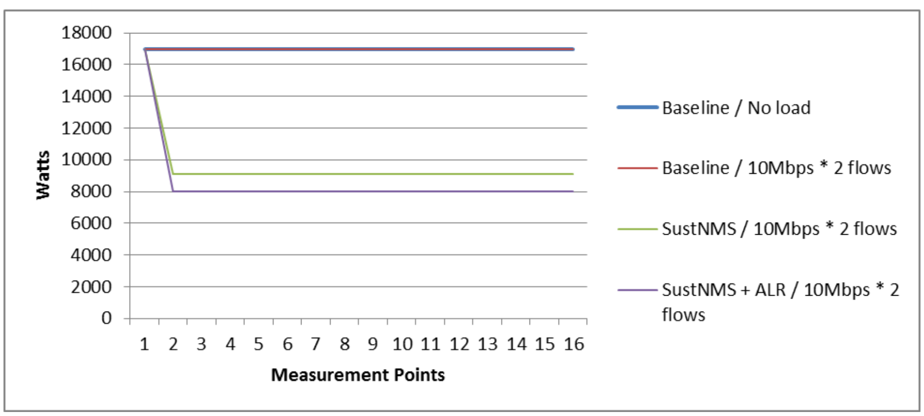

Figure 10: Results for a not load proportional scenario

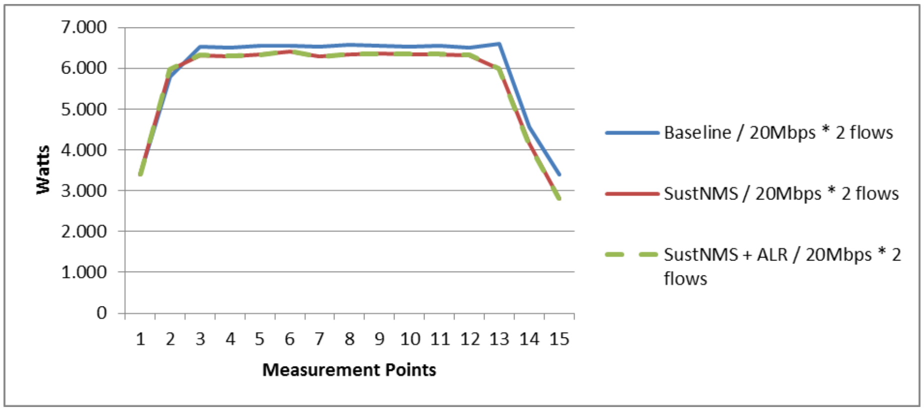

Figure 11: Results for a high load scenario with load proportional power profiles

6.3. Validating the use of decision trees

To validate the tree decisions, we compared the results from the Utility Function module, which used the predictions from the Analytical Solvers, with the results while running the experiments in the emulation environment. From the Utility Function results, when both flows are smaller than 12 Mbps, the expected capabilities combination (the one with the highest Utility Function grade) is SC with ALR. If one of them is bigger than 12 Mbps, SustNMS-Sustainability is the expected selection, with ALR being applied if one of the flows is smaller than 10Mbps. When the sum of both flows surpasses 30 Mbps, the maximum each node can handle, node 14 will lose many packages. In this case, the Utility Function ended up selecting SustNMS-Performance that is, it is better to wake up some additional devices and save less, in order to loose fewer packages, ensuring more QoS.

Considering the expected decision tree behavior, we performed the following experiments in the testbed: (1) Flow 1: 2Mbps, Flow 2: 2Mbps; (2) Flow 1: 14Mbps, Flow 2: 6Mbps; (3) Flow 1: 15Mbps, Flow 2: 15Mbps; (4) Flow 1: 24Mbps, Flow 2: 24Mbps. Note that none of them is equal to the workloads depicted in Table 6.3, so that we can check the Scikit tool interpolation. Table 6.3 details what happened for each case. All the experiments had the result as expected.

Flow 1 / Flow 2	**Expected from the Utility Function**	**Happened in the Emulation environment**
2Mbps / 2Mbps	SC + ALR in both flows	SC + ALR in both flows
14Mbps / 6Mbps	SustNMS-Sustainability + ALR in the flow with 14Mbps, only SustNMS-Sustainability in the flow with 6Mbps	SustNMS-Sustainability + ALR in the flow with 14Mbps, only SustNMS-Sustainability in the flow with 6Mbps
15Mbps / 15Mbps	SustNMS-Sustainability in both flows	SustNMS-Sustainability in both flows
24Mbps / 24Mbps	SustNMS-Performance in both flows	SustNMS-Performance in both flows

Table 4: Method validation

6.4. Checking Dinamicity

When the method is started, an operational policy is translated, giving the network level information such as the environment conditions (save energy in any condition). If the operational policy states, for instance, that the method will only save energy if the network load is less than 50%, the random workloads are generated only for cases in which the load is smaller than 50% the maximum load the network can handle.

Other terms that are refined are the period of the day the energy efficiency functionalities will take place (during the night and during the day), and which capabilities are allowed in each case. For instance, during the night, all capabilities are allowed, while, during the day, only link rating capabilities are allowed, as exemplified in Section 4. This can be the determined because the link rating capabilities enforcement are in hundreds of milliseconds order of magnitude while sleeping can take seconds to be fully operational.

Figures 12 and 13 illustrate our graphical user interface during the night, applying different functionalities, and during the day, applying only ALR. It is also important to note that ALR is only applicable when the workload is smaller

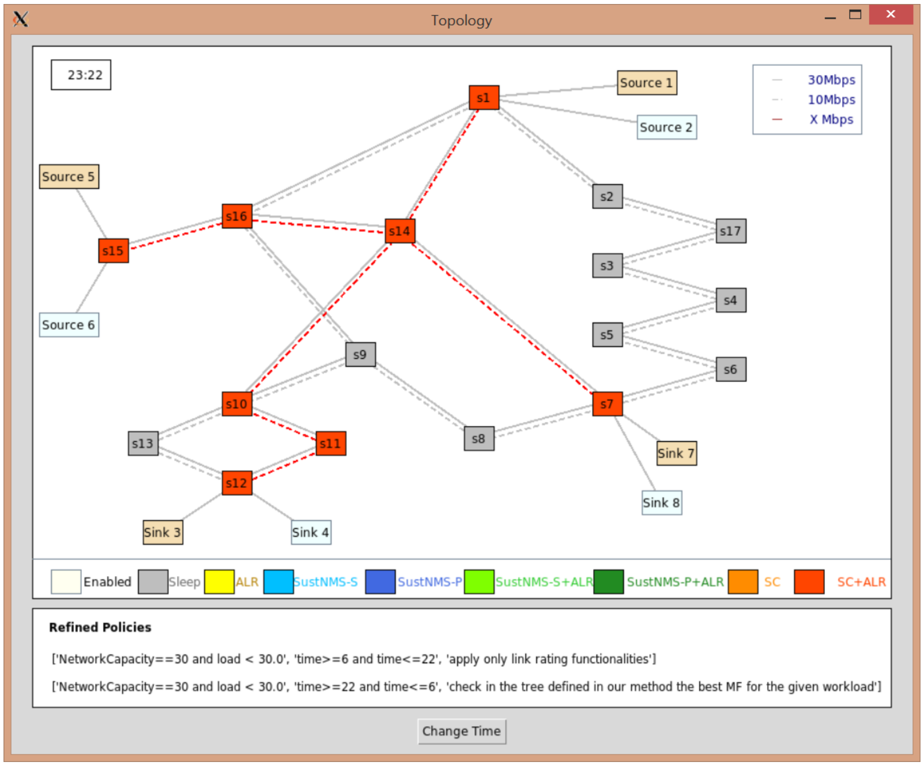

Figure 12: Operation during the night

than 10Mbps.

7. Discussions

Recently, different approaches to improve energy efficiency have been proposed. However, such energy management capabilities were designed to operate autonomously and independently from each other. Besides operating the network in a conflict-free manner, the orchestration of energy efficiency capabilities performed by the SOS method allows to combine more than one capability at the same time in the nodes. Such coordination leads to bigger savings than enabling just one capability, as demonstrated by the experiments. Besides, the method can choose which capability (-ties) is (are) the best for a given scenario, thus turning on the one that has the best results. Before SOS, there was no available

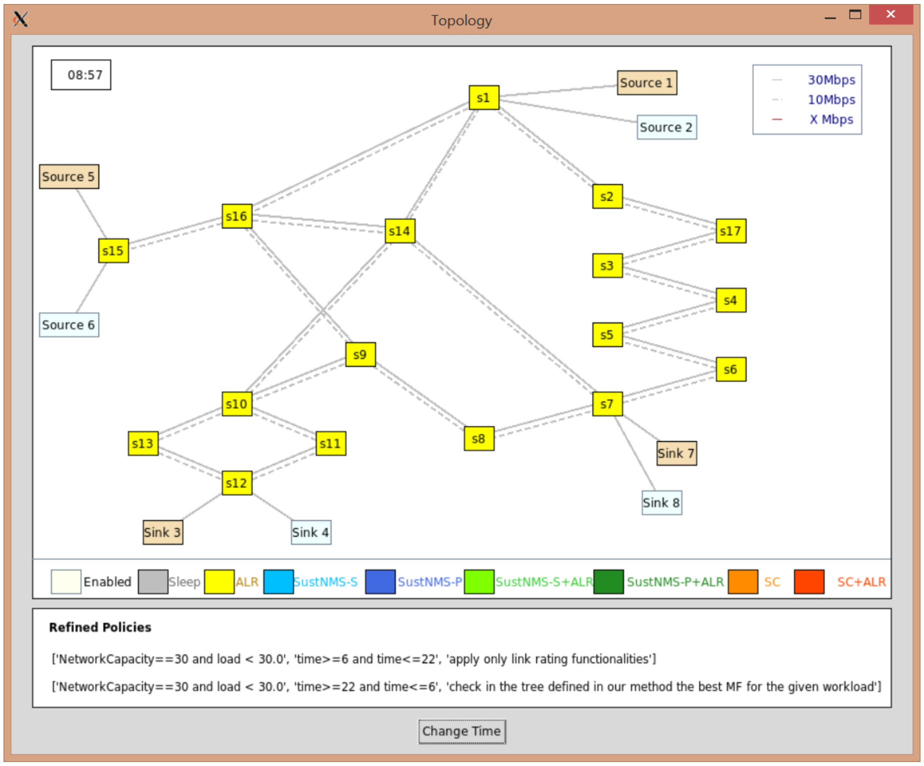

Figure 13: Operation during the day

method that could orchestrate or coordinate energy efficiency capabilities, nor consider business directives in such operation in an automated way.

The existing refinement methods focus on translating policies from a high to a lower level of abstraction, supporting policy analysis, resources discovery, and dynamicity. Despite the possibility of extending them in order to comprise energy efficiency capabilities translation, such methods do not perform energy efficiency capabilities orchestration. That is, they are not able to combine more than one capability to save more energy, nor to ensure a conflict-free operation. They are not able to do such orchestration for other domains either, such as QoS or security, which can be one of the SOS method extensions for deployment in production networks, with more than one capability type.

In order to tackle scalability, the decision tree of SOS is constructed offline, that is, without affecting the network operation. Only after the tree is built, the tree is deployed in the network to provide the best combination of capabilities given a network state (flows, usage) in a conflict-free operation.

Summarizing, the method has the following advantages: (a) coordination of energy efficiency capabilities allowing the operator to optimize the energy consumption; (b) besides the possibility of saving more energy, the orchestration ensures a conflict-free operation. A conflicting operation could lead to undesired behavior, failures, and, consequently, reduced quality of service. Besides, applying a capability not suited to the current bandwidth utilization value might lead to congestion or packet losses; (c) business-level directives, refined down to the device and instance policy levels, in an automated way, bring high-level goals to the network operation. Such automation turns the management task easier, less manual, and less prone to errors.

8. Conclusions and Future Works

Energy efficiency functionalities help operators and service providers to reduce operational costs and GHG emissions in their ICT infrastructures. Sustainability-oriented network management policies can help by bringing business directives into the network, turning the management more automated and less error prone, also reducing operational costs. In this regard, we proposed a method able to orchestrate energy efficiency capabilities considering sustainability-oriented policies refinement, enabling a more energy efficient and automated infrastructure. To the best of the author's knowledge, this is the first work that comprises the complete refinement of such policies including the orchestration of energy efficiency functionalities. The proposed method was validated using different experiments, testing the Utility Function, checking the extra savings when combining more than one capability, the decision tree interpolation and dynamicity aspects.

As future works, other refinement techniques can be automated and incorporated in the SOS method. Other metrics or constraints, such as reliability, could be directly incorporated in the policies. Besides, the scenario dynamicity must be further developed. The method could also be expanded to support refinement including orchestration of QoS and security capabilities. Or even be applied in other scenarios, such as a cloud environment, to orchestrate energy efficiency capabilities not only for networks, but also for other parts, such as computing.

Acknowledgments

Project funded by Ericsson Telecomunicações S.A., Brazil.

References

[1] Ericsson, Ericsson Energy and Carbon Report - On the Impact of the Networked Society, Tech. rep., Ericsson (July 2013).
URL `http://www.ericsson.com/res/docs/2013/ericsson-energy-and-carbon-report.pdf`

[2] J. Malmodin, s. Moberg, D. Lundn, G. Finnveden, N. Lvehagen, Greenhouse gas emissions and operational electricity use in the ict and entertainment & media sectors, Journal of Industrial Ecology 14 (5) (2010) 770–790.

[3] J. Malmodin, P. Bergmark, D. Lundn, The future carbon footprint of the ict and e&m sectors, in: ICT4S 2013. Proceedings of the First International Conference on Information and Communication Technologies for Sustainability, ETH Zurich, 2013, pp. 12–20.

[4] P. Jppinen, R. Guarneri, L. M. Correia, An applications perspective into the future internet, Journal of Network and Computer Applications 36 (1) (2013) 249 – 254. `doi:http://dx.doi.org/10.1016/j.jnca.2012.08.009`.
URL `http://www.sciencedirect.com/science/article/pii/S1084804512001853`

[5] G. Cook, T. Dowdall, D. Pomerantz, Y. Wang, Clicking Clean: How Companies are Creating the Green Internet, Tech. rep., Greenpeace (April 2014).
URL http://www.greenpeace.org/usa/Global/usa/planet3/PDFs/clickingclean.pdf

[6] GeSI, SMARTer 2020: The Role of ICT in Driving a Sustainable Future, Tech. rep., The Climate Group on behalf of the Global eSustainability Initiative (GeSI) (2012).
URL http://gesi.org/SMARTer2020

[7] Emerson Electric Co., Energy Logic: Reducing Data Center Energy Consumption (2009).
URL http://www.cisco.com/web/partners/downloads/765/other/Energy_Logic_Reducing_Data_Center_Energy_Consumption.pdf

[8] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, H. Liu, Energy proportional datacenter networks, in: Proceedings of the 37th Annual International Symposium on Computer Architecture, ISCA '10, ACM, New York, NY, USA, 2010, pp. 338–347.

[9] D. Kliazovich, P. Bouvry, S. Khan, DENS: Data center energy-efficient network-aware scheduling, in: Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Cyber, Physical and Social Computing (CPSCom), 2010, pp. 69–75.

[10] G. Koutitas, L. Tassiulas, I. Vlahavas, Energy efficiency monitoring in data centers: Case study at international hellenic university, UTH and IHU (CI)-WP4 (2012).
URL http://www.fp7-trend.eu/system/files/content-public/248-trend-friends-workshop-gent-14-february-2012-presentations/wp4energyefficiencymonitoringinsmallscaledatcentersuthkoutitas.pdf

[11] C. Kachris, I. Tomkos, Power consumption evaluation of all-optical data center networks, Cluster Computing 16 (3) (2013) 611–623.

[12] Verizon, 2013 Corporate Responsibility Supplement, Tech. rep., Verizon (2013).
URL http://responsibility.verizon.com/assets/docs/cr-report-supplement-2013.pdf

[13] C. Gunaratne, K. Christensen, B. Nordman, S. Suen, Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR), Computers, IEEE Transactions on 57 (4) (2008) 448–461.

[14] M. Mostowfi, K. Christensen, Saving energy in LAN switches: New methods of packet coalescing for Energy Efficient Ethernet, in: IGCC'11, 2011, pp. 1–8.

[15] A. Cianfrani, V. Eramo, M. Listanti, M. Polverini, A. Vasilakos, An ospf-integrated routing strategy for qos-aware energy saving in ip backbone networks, Network and Service Management, IEEE Transactions on 9 (3) (2012) 254–267.

[16] M. Zhang, C. Yi, B. Liu, B. Zhang, GreenTE: Power-aware traffic engineering, in: Network Protocols (ICNP), 2010 18th IEEE International Conference on, IEEE, 2010, pp. 21–30.

[17] C. H. Costa, M. C. Amaral, G. C. Januario, T. C. Carvalho, C. Meirosu, SustNMS: Towards service oriented policy-based network management for energy-efficiency, in: Sustainable Internet and ICT for Sustainability (SustainIT), 2012, 2012, pp. 1–5.

[18] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, N. McKeown, Elastictree: Saving energy in data center networks, in: NSDI'10, 2010, pp. 17–17.

[19] R. Craven, J. Lobo, E. Lupu, A. Russo, M. Sloman, Policy refinement: decomposition and operationalization for dynamic domains, in: Network

and Service Management (CNSM), 2011 7th International Conference on, IEEE, 2011, pp. 1–9.

[20] A. Riekstin, G. Januario, B. Rodrigues, V. Nascimento, T. Carvalho, C. Meirosu, A Survey of Policy Refinement Methods as a Support for Sustainable Networks, submitted to IEEE Communications Surveys and Tutorials.

[21] A. Riekstin, G. Januario, B. Rodrigues, V. Nascimento, M. Pirlea, T. Carvalho, C. Meirosu, Orchestration of energy efficiency functionalities for a sustainable network management, in: Network Computing and Applications (NCA), 2014 IEEE 13th International Symposium on, 2014, pp. 157–161.

[22] R. Schlenk, C. Lange, H. Lehmann, R. Vleugel, Taxonomy of dynamic power saving techniques in fixed broadband networks, in: Photonic Networks, 14. 2013 ITG Symposium., 2013, pp. 1–8.

[23] R. Bolla, R. Bruschi, A. Ranieri, Performance and power consumption modeling for green COTS software router, in: Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International, IEEE, 2009, pp. 1–8.

[24] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, J. Maestro, IEEE 802.3az: the road to energy efficient ethernet, Communications Magazine, IEEE 48 (11) (2010) 50 –56.

[25] J. Strassner, Policy-Based Network Management: Solutions for the Next Generation, 1st Edition, Morgan Kaufmann, 2003.

[26] T. C. M. B. Carvalho, A. C. Riekstin, M. Amaral, C. H. A. Costa, G. C. Januario, C. K. Dominicini, C. Meirosu, Towards Sustainable Networks - Energy Efficiency Policy from Business to Device Instance Levels, in: ICEIS, SciTePress, 2012, pp. 238–243.

[27] D. C. Verma, Policy-based networking: architecture and algorithms, New Riders Publishing, 2000.

[28] A. K. Bandara, E. C. Lupu, A. Russo, Using event calculus to formalise policy specification and analysis, in: Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on, IEEE, 2003, pp. 26–39.

[29] J. Rubio-Loyola, Towards the Policy Refinement Problem in Policy-based Management Systems: A synthesis study, VDM Publishing, 2008. URL `http://books.google.com/books?id=hOOHNQAACAAJ`

[30] K. Twidle, E. Lupu, N. Dulay, M. Sloman, Ponder2-a policy environment for autonomous pervasive systems, in: Policies for Distributed Systems and Networks, 2008. POLICY 2008. IEEE Workshop on, IEEE, 2008, pp. 245–246.

[31] R. Craven, J. Lobo, J. Ma, A. Russo, E. Lupu, A. Bandara, Expressive policy analysis with enhanced system dynamicity, in: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, ASIACCS '09, ACM, New York, NY, USA, 2009, pp. 239–250.

[32] A. Bakshi, A. Talaei-Khoei, P. Ray, Adaptive policy framework: A systematic review, Journal of Network and Computer Applications 36 (4) (2013) 1261 – 1271. `doi:http://dx.doi.org/10.1016/j.jnca.2012.12.007`. URL `http://www.sciencedirect.com/science/article/pii/S1084804512002548`

[33] V. Nascimento, B. Rodrigues, A. Riekstin, T. Carvalho, C. Meirosu, Sustainability Information Models for Energy Efficiency Policies, submitted to Sustainable Internet and ICT for Sustainability (SustainIT), 2015.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn:

Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[35] G. C. Januario, C. H. Costa, M. C. Amarai, A. C. Riekstin, T. C. Carvalho, C. Meirosu, Evaluation of a policy-based network management system for energy-efficiency, in: Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on, IEEE, 2013, pp. 596–602.

[36] M. Ricca, A. Francini, S. Fortune, T. Klein, An assessment of power-load proportionality in network systems, in: Sustainable Internet and ICT for Sustainability (SustainIT), 2013, 2013, pp. 1–8.

[37] S. Ricciardi, D. Careglio, U. Fiore, F. Palmieri, G. Santos-Boada, J. Solé-Pareta, Analyzing local strategies for energy-efficient networking, in: Proceedings of the IFIP TC 6th International Conference on Networking, NETWORKING'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 291–300. URL http://dl.acm.org/citation.cfm?id=2039912.2039944

[38] B. Rodrigues, A. Riekstin, G. Januario, V. Nascimento, T. Carvalho, C. Meirosu, GreenSDN: Bringing Energy Efficiency to an SDN Emulation Environment, in: Integrated Network and Service Management (IM), 2015 14th IFIP/IEEE Symposium on, 2015, p. to appear.