A semantic enhanced Power Budget Calculator for distributed computing using IEEE 802.3az

Zhu, H.; van der Veldt, K.; Zhao, Z.; Grosso, P.; Pavlov, D.; Soeurt, J.; Liao, X.; de Laat, C.

DOI
10.1007/s10586-014-0395-7
Publication date
2015
Document Version
Final published version
Published in
Cluster Computing

Citation for published version (APA):
Zhu, H., van der Veldt, K., Zhao, Z., Grosso, P., Pavlov, D., Soeurt, J., Liao, X., & de Laat, C. (2015). A semantic enhanced Power Budget Calculator for distributed computing using IEEE 802.3az. *Cluster Computing*, *18*(1), 61-77. https://doi.org/10.1007/s10586-014-0395-7

A semantic enhanced Power Budget Calculator for distributed computing using IEEE 802.3az

Hao Zhu, Karel van der Veldt, Zhiming Zhao, Paola Grosso, Dimitar Pavlov, Joris Soeurt, Xiangke Liao & Cees de Laat

Cluster Computing
The Journal of Networks, Software Tools and Applications

ISSN 1386-7857
Volume 18
Number 1

Cluster Comput (2015) 18:61-77
DOI 10.1007/s10586-014-0395-7

Springer

Cluster Comput (2015) 18:61–77
DOI 10.1007/s10586-014-0395-7

A semantic enhanced Power Budget Calculator for distributed computing using IEEE 802.3az

Hao Zhu · Karel van der Veldt · Zhiming Zhao · Paola Grosso · Dimitar Pavlov · Joris Soeurt · Xiangke Liao · Cees de Laat

Received: 24 January 2013 / Revised: 22 September 2013 / Accepted: 17 July 2014 / Published online: 15 August 2014

Abstract Energy efficiency is becoming an important requirement in more and more computing systems for optimizing resource allocation and task scheduling. By switching active copper Ethernet links to a low power model the IEEE 802.3az protocol can reduce the network energy consumption when no traffic exists. However, the effect of 802.3az heavily depends on network traffic patterns, which makes its utilization challenging in scheduling computing tasks. In this research, we examined the 802.3az technology with the goal of deploying it in distributed computing systems such as clusters. We devised an energy budget calculator that includes the energy model of 802.3az compliant Ethernet devices and supports the resource management service. We show a few practical examples of how applications can better plan their execution by integrating this knowledge in their decision strategies. We also present a solution for enhancing the calculator by using a semantic energy information system.

H. Zhu · K. van der Veldt · Z. Zhao (✉) · P. Grosso · C. de Laat · D. Pavlov · J. Soeurt
University of Amsterdam, Science Park 904, Amsterdam, Netherlands
e-mail: z.zhao@uva.nl

H. Zhu
e-mail: h.zhu@uva.nl

K. van der Veldt
e-mail: karel.vd.veldt@uva.nl

P. Grosso
e-mail: p.grosso@uva.nl

C. de Laat
e-mail: delaat@uva.nl

X. Liao
National University of Defense Technology, Changsha, China
e-mail: xkliao@nudt.edu.cn

Keywords IEEE 802.3az · Distributed computing · Energy Efficient Ethernet · Power Budget Calculator

1 Introduction

Many scientific applications require the management of large volumes of data in distributed computing environments [1]. For instance, the processing of experimental data in the astronomy and high energy physics domains often requires resources in many locations and the movement of terabytes or petabytes of data. These intensive data transfer characteristics obviously lead to huge energy consumption on networking devices. The pursuit of solutions that optimize the efficiency of networking equipment, along with advances in energy efficiency concepts, have led to the development of several techniques, which contribute to energy savings. These techniques include powering down switch ports when unused [2], adjusting transmission power based on cable length [2], and switching active switch ports into sleep mode when no traffic is expected [3]. The first two methods are part of what is called informally by some vendors *Green Ethernet*; the latter one becomes the core technology of the Energy Efficient Ethernet (EEE), and it has recently been standardized as the IEEE 802.3az protocol in 2010.

Instead of shutting down unused ports, 802.3az provides flexible mechanisms for customizing device energy consumption based on network traffic. It offers applications the opportunity to consider its energy consumption by tuning its communication loads while scheduling computing tasks; however, it also requires profound understanding of 802.3az energy behaviour on different network patterns. Although the 802.3az standard has been finalized in 2010, compliant products have only appeared in the market recently. Vendors such as Cisco and Huawei are incorporating 802.3az into high end switches which can be used in clusters. Therefore, understanding the actual energy model of 802.3az compliant devices is crucial to construct device specific profiles for the resource brokers of the infrastructure to provide most suitable resources for data intensive applications.

In [4], we did initial investigation on the energy behaviour of 802.3az switches under different scenarios and using different traffic patterns. In this paper, we extend this initial work and give a detailed discussion on an energy budget calculator to estimate the power consumption of the devices based on the input communication load. The paper is organized as follows. First, we give an overview of the 802.3az protocol. After that, we present an experimental study of two selected 802.3az devices. Finally, we discuss a prototype of an energy budget calculator service and present its integration with a semantic information system.

2 Link sleep technologies and Energy Efficient Ethernet

Link sleep technologies are the basis of the 802.3az standard: ports in a device or the whole device are allowed to switch into sleep mode for reducing energy consumption during periods of idle or low utilisation. In this section, we will give an overview of the link sleep technologies and the 802.3az standard.

2.1 Link sleep technologies

In 2003, Gupta et al. [5] noticed the huge energy consumption of networking devices in the Internet and argued the benefit of using network protocols to switch the networking devices into sleep or low power mode. Their following work [6] demonstrated the feasibility of increasing low-power modes for switches in a campus network environment. The basic strategy is to let the network interfaces sleep if the inter-packet arrival delays are larger than the transition time. However, the inter-packet arrival delays were difficult to predict; if the actual delay was not long enough, the energy saved by sleep could be less than the energy caused by the state transitions. A better approach was presented in later work [7]: the interface would only go into sleep when the buffer occupancy was lower than a predefined threshold. This approach could reduce the number of transitions but it might also cause high delays for the packets that were already in the buffer before device in the sleep state. To address this problem, Herreira-Alonso et al. [8] proposed to switch the link into sleep only when the buffer is empty. Ananthanarayanan et al. [9] investigated these problems from two aspects: waking up port when traffic comes namely Wake-on-Packet and receiving ingress packets on behalf of sleeping ports namely Shadow Port.

2.2 EEE protocol

As a signaling protocol, 802.3az allows the transmitter to notify the receiver that a gap in data transmission is imminent and the switch port can thus go into low power mode.

The 802.3az protocol distinguishes five main states of a switch port:

1. *Active*, when the port is in fully operational state;
2. *Quiet*, the port is in lower power mode;
3. *Sleep*, the link is powering down and transitioning to the Quiet state;
4. *Refresh*, the link is being refreshed;
5. *Wakeup*, the link is powering on and transitioning to the Active state.

The energy is saved when the port is in the *Quiet* state.

The state transition is achieved by sending a special type of Ethernet control code – a *LPI* code, instead of normal IDLE control code. After a predefined period $T_s = time\ to\ sleep$ expires, the link becomes completely quiescent for a period $T_q = time\ quiescent$ until a *refresh* signal is sent during the period $T_r = time\ refresh$. This *refresh* signal is sent with a different frequency, depending on the linkspeed; it allows the detection of unplugged or failed links. To resume data transmission, the transmitter sends a normal IDLE control code, after which the receiver wakes up and can receive data. The transmitter waits for a period of time $T_w = time\ to\ wake$ before it starts transmitting. Any data ready to be sent is buffered so that no data is lost. The 802.3az protocol predefines the value of T_w, which is generally equal the time it takes to send one maximum length packet for the link speed used on the link. T_d is the total active time of the port. Figure 1 shows an illustration of these state transitions. The 802.3az protocol works only when the devices at both ends of the link advertise this capability.

As 802.3az compliant devices have only been released in the market recently, most 802.3az energy studies are based on simulation results. For instance, Reviriego et al in 2009 [10] first studied and explained that state transitions in 802.3az caused energy consumption and decreased the energy efficiency of the switch when compared to the ideal situation. The 802.3az standard defines the signaling between connected network devices, but the protocol does not specify a power-saving policy; i.e. when an interface should go into power saving mode and the predefined values of T_s and T_w. This implies the energy consumption of 802.3az compliant devices depends on the specific vendor implementation. Performing a practical study of newly available switches and deriving their energy profile is therefore essential for applications that want to use network resources in an efficient manner.

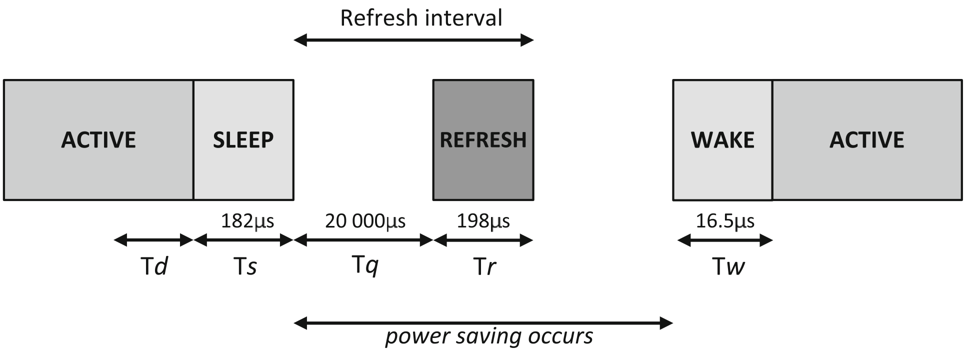


Fig. 1 Illustration of 802.3az state transitions (timing for 1000BASE-T link speed)

Thus, our main motivation is to investigate the practical issues in using 802.3az in scheduling of distributed applications to achieve system level energy efficiency. More specifically, we:

1. Investigate the energy behaviour of 802.3az devices in a real network situation;
2. Model the energy characteristics of 802.3az compliant devices;
3. Provide solutions for applying this energy model in a distributed computing environment.

3 Energy characteristics of 802.3az compliant devices

We investigate the energy characteristics of 802.3az-enabled switches for different values of the link speed, the transmission rate, the transport protocol and the inter-frame delay. We use two different 802.3az-enabled switches: Cisco SG300-28 (HW v2, SW v1.1.2.0) [11], and Huawei S1728GWR-4 [12]. The testbed consists of these switches and several test servers. Each test server is equipped with a dual port Intel I350-T2 NIC [13]. The power consumption is measured using a Racktivity ES6024-16 *PDU*, which provides power consumption measurement once per second.

The switch under investigation is connected to two test servers acting as the traffic source and destination. To simulate traffic over all ports, the remaining ports are looped in a daisy chain fashion, so that all ports draw power. Each group of ports is assigned a unique VLAN to force traffic to use the external path, instead of the switch backplane. In this way, all ports receive the same traffic and have the same state transitions.

3.1 Maximum energy savings

We first measured the power consumption of the switches where there is no traffic at all (idle), and later on when the bandwidth on all ports is saturated. When 802.3az is enabled, the Cisco switch consumes 11W when idle and at most 19W when saturated, while the Huawei switch consumes at least 15W and at most 22W in the two cases. When 802.3az is disabled, both switches consume the maximum amount constantly regardless of traffic. This experiment shows a difference of about 7W (about 32 %) for the Huawei switch and 8W (about 42 %) for the Cisco switch when using maximum link load and no link load. Both switches show a clear decrease in power consumption when the 802.3az protocol is enabled and the link speed of all ports is 1Gbps.

3.2 Energy consumption versus throughput

Our second experiment determines the energy consumption of the 802.3az switches using different transmission rates. Our goal is to determine the relationship between energy consumption of these 802.3az-enabled switches and throughput.

We generated traffic from the source to the destination server in our test setup. We performed multiple test runs, setting the desired transmission rate at each run. To limit the transmission rate, we used both the built-in functionality of Iperf, as well as the `tc` Linux traffic control tool. Both methods were used since Iperf only supports traffic rate limiting of *UDP* traffic. However, there were no clear differences in the results between these two methods. The results of this experiment when using 1Gbit link speed and *TCP* traffic are given in Figs. 2 and 3. The results from the same test performed with *UDP* traffic are given in Figs. 4 and 5.

When using TCP traffic, both switches show a linear increase in power consumption to maximum power consumption at around 450 Mbps for the Huawei switch and 400 Mbps for the Cisco switch. However, when using UDP traffic, both switches consume maximum energy except when the throughput exceeds 50 Mbps. This may be explained by the fact that, in comparison to TCP, UDP is a stateless protocol and transmits without flow control. Lack of flow optimization could decrease the idle time interval so that the interface could hardly go to sleep at all.

Overall, both devices confirm the simulations performed by earlier research [14]. The difference in the ”tipping point” between the two devices—the point at which the switches start consuming maximum energy, might be explained by a

Fig. 2 TCP, 1000BASE-T Huawei S1728GWR-4P

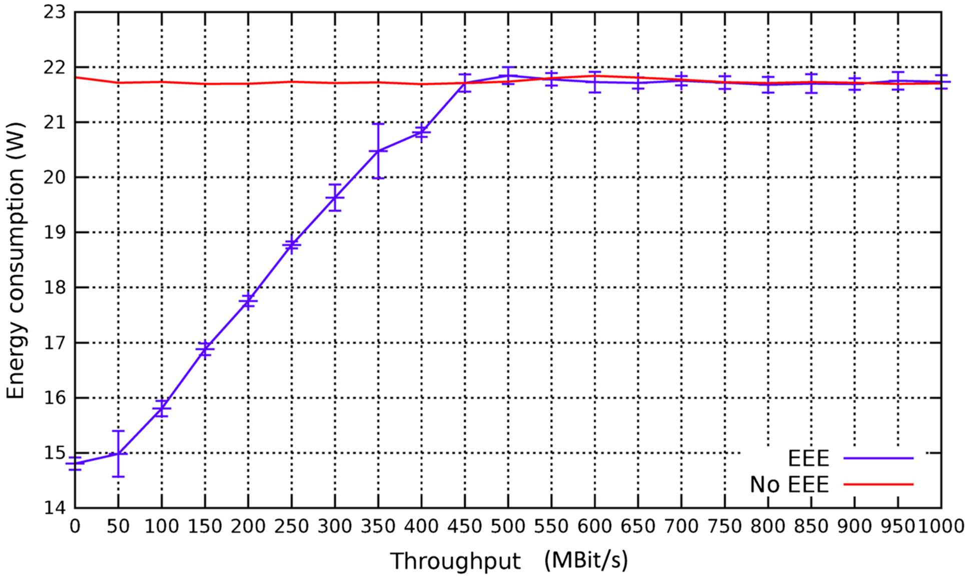
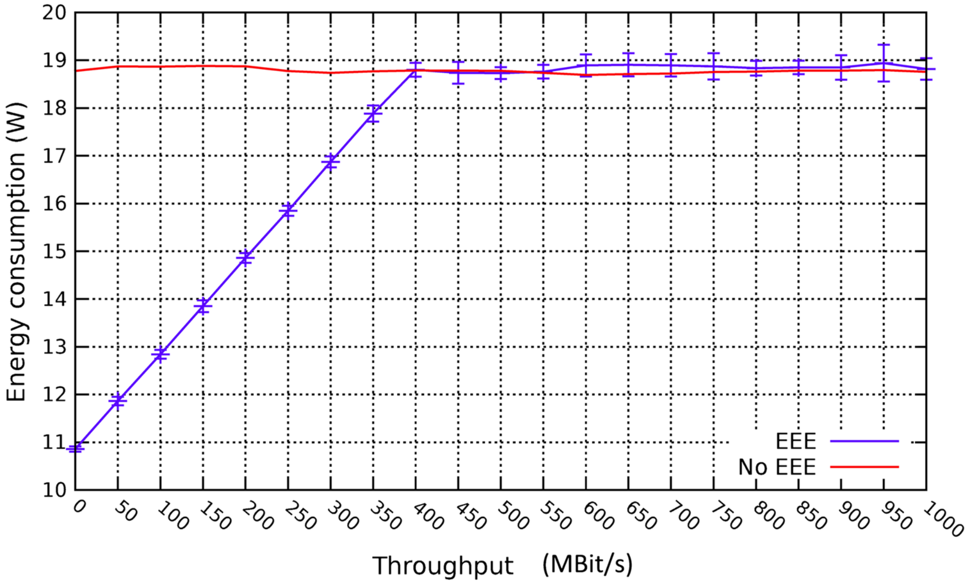


Fig. 3 TCP, 1000BASE-T Cisco SG300-28



difference in the implemented T_s value for both switches. A different T_s value would influence the respective switch's power consumption with the same traffic pattern.

Also, when observing the pattern of increase in the power consumption, we see a linear increase instead of the logarithmic increase expected. Although we believe this is related to the specific vendor implementation, we were unable to verify this, and believe this should be examined further as part of future research.

3.3 Energy consumption versus inter-frame delay

The last experiment measures the energy consumption of an 802.3az-enabled switch and relates it to the inter-frame delay of a stream of consecutive frames. The purpose of this experiment is twofold. The time to enter and exit the lower power mode is not formally defined in the protocol specification; rather, it is vendor- and implementation-specific. The first goal of this experiment is to determine whether it is possible to experimentally retrieve timing information using our test setups. The second goal is to try and find an explanation for the lack of decrease in energy consumption when using *UDP* traffic and 802.3az-enabled switches.

We used the `mz` packet generation tool [15] to send empty frames (just the frame header) at specific time intervals. We sent frames with different inter-frame delays. Each test ran for one minute and energy consumption was averaged over that period. The results of this experiment are given in Figs. 6 and 7. The plots show the average energy consumption of the

Fig. 4 UDP, 1000BASE-T Huawei S1728GWR-4P

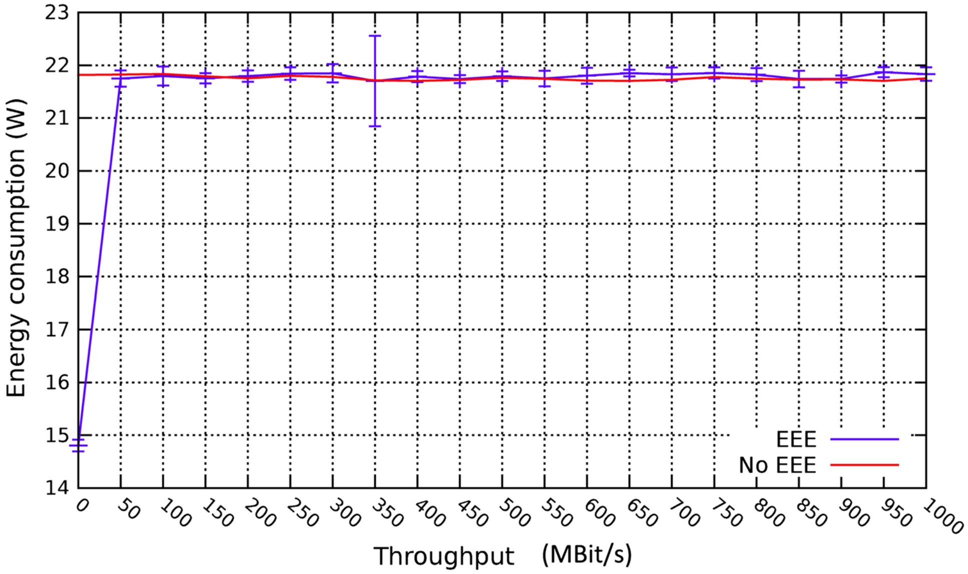
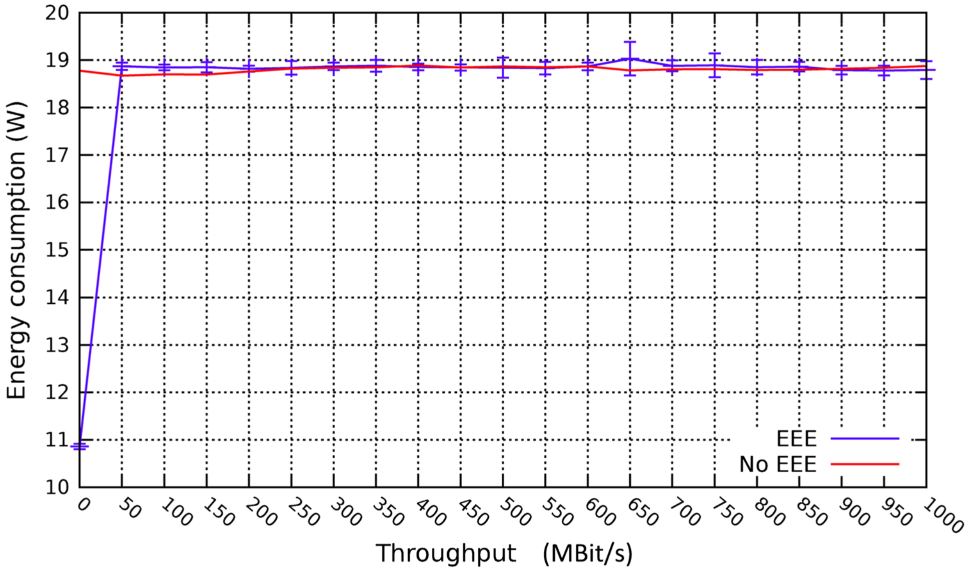


Fig. 5 UDP, 1000BASE-T Cisco SG300-28



switch per delay interval, and, thus, the standard deviation is given as well.

The results of this experiment show a clear relation between the inter-frame delay and the switch energy consumption at 1Gbps link speed (see Figs. 6 and 7). We note that the Huawei device starts saving energy at smaller inter-frame delay values than the Cisco device. We can relate this to the fact that the Huawei switch saves energy at a higher throughput that the Cisco switch (see Sect. 3.2).

Although a clear relation between the frame pattern and the energy consumption can be derived from these test results, the reported timing values do not seem to match the values defined in the 802.3az standard. To be able to save power in low power mode, the interface has to be idle for a period equal to at least $min(T_s)+min(T_w)$ (see Fig. 1), which equals 198.5μ for 1000BASE-T. This partly explains our test results, which show a decrease in power consumption for an inter-frame delay of higher than 300μ. The possible reason of a power decrease for an inter-frame delay of lower than 200μ is the vendor-specific implementations of 802.3az.

3.4 Summary

The results of the 802.3az experiments generally confirm the theoretical analysis of the IEEE 802.3az protocol, performed by previous research [10,16]. The most significant difference we observed is the fact that the amount of consumed power increases linearly with the increase in transmission speed, while previous research suggests that this amount should increase logarithmically. Furthermore, instead of increasing steadily until it reaches the maximum transmission speed, we observed that power usage increases until it hits a tipping

Fig. 6 Frame pattern Huawei S1728GWR-4P

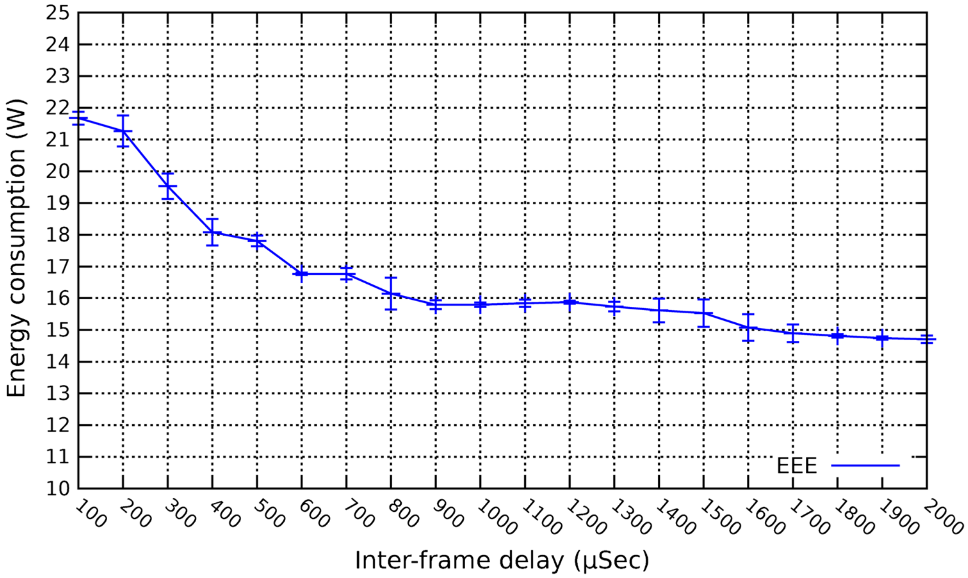
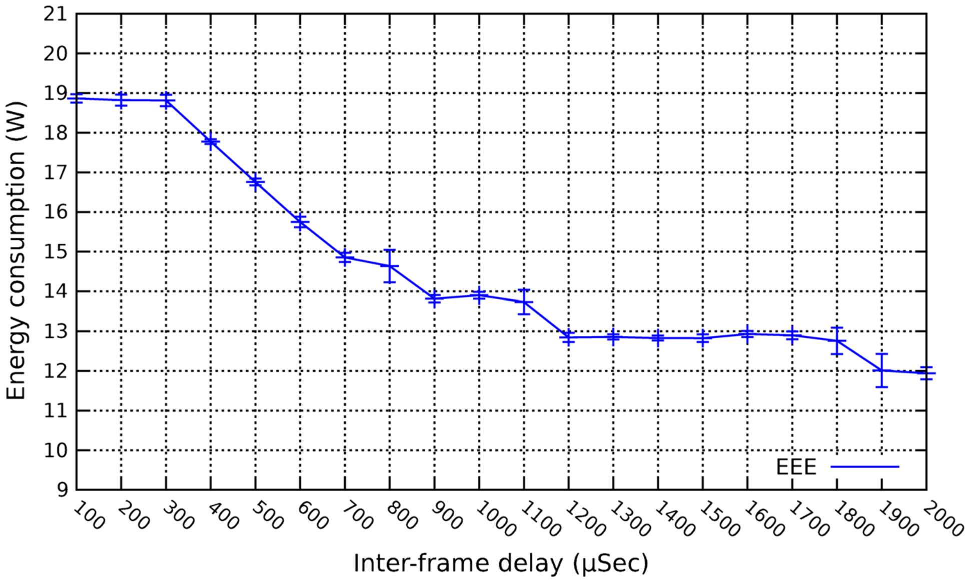


Fig. 7 Frame pattern Cisco SG300-28



point, after which the consumption saturates and stabilizes. We believe that this is related to the protocol's state transition timers, which prevent the networking controller from falling asleep after a certain boundary. As these timers are also dependent on the specific vendor implementation, we could not look further into this matter. This effect can clearly be seen in Fig. 2 and 3.

Moreover, when using UDP traffic, we could not observe a clear dependency between throughput and device energy consumption. This could be explained by the nature of the UDP protocol. In comparison, the TCP protocol has to wait for acknowledgements of the packets sent, while UDP does not, since it is a stateless protocol, and transmits as fast as the rate limiting allows it. Another difference is that UDP lacks flow control like Nagle's algorithm [17], which combines a number of outgoing messages, and sends them all at once. While this could lead to less time for the interface going into low power mode, we believe that this behaviour is odd and should be examined further by future research.

Finally, in the experiments on the 802.3az sleep timers by varying the inter-frame delay, the results show a decrease in energy consumption when increasing the inter-frame delay. The relation is neither linear, nor exponential (as simulated in previous research [14]).

4 Power Budget Calculator prototype

From the experiments, we observed the differences in the implementations of different vendors. Their overall behaviours are similar, but the amount of energy saved and the threshold of effectiveness of the protocol differ.

Using the results presented in Sect. 3, we devised a *Power Budget Calculator*[1] service for estimating power usage in clusters equipped with 802.3az-enabled switches. Our calculator covers three scenarios: (1) fixed throughput as a function of the number of active ports, (2) fixed amount of data transfer over a fixed number of active ports as a function of throughput and (3) time distribution for a task as a function of the active computing nodes.

Combining these three scenarios, we can estimate the total power usage of a switch in two situations: as a function of the number of computing nodes and the time needed for a task's completion (*Task-based Estimate*, see Sect. 4.1); and as a function of the switch throughput when the amount of data to be transmitted is fixed (*Data-based Estimate*, see Sect. 4.2).

We have implemented the task-based power usage estimation separately from the fixed-time estimation of bandwidth.

4.1 Task-based estimate

Using task-based estimation, the PBC can operate in the context of parallel computing and output a recommendation on a power-optimal parallelization of a certain task. The time for completing a task decreases while the number of utilized computing nodes increases (Formula 1).

$$T_t = \frac{T_c}{n} + (n-1) * C \tag{1}$$

where T_c is the time needed to complete the task if only by *one node*; (T_t) is the total *time* needed to complete the task if in parallel execution and C is the communication time cost. This is a simple time model for parallel computing.

The related parameters of the PBC are: the number of available nodes, the switch's profile, usable ports per switch, total time to complete the task on one node, communication overhead time, network transmission speeds, and average power consumption per node.

In this model, we use an average power consumption of computing node as input and perform calculations with that. In our previous work we have started to evaluate the energy profile of computing nodes [18] and we plan to include this in a future version of the calculator.

By analyzing the input parameters and performing time distribution estimation, we estimate the total energy usage of the switch for a particular task as a function of the number of used ports on a switch.

Figures 8 and 9 show example output of the task-based estimation tool using the parameters shown in Table 1.

As can be seen in the Fig. 8, the power usage of the switches follows a hyperbolic curve, as the energy consumption of the devices is heavily dependent on the time needed for the execution of a task. Note the jump in power consumption, which corresponds to adding a second switch to handle the needed number of nodes. The total power consumption (shown in Fig. 9) follows a similar shape. When the computing nodes' power consumption is included, the energy consumption grows quite fast in comparison to the consumption of the switches.

We can see from these results that using eight compute nodes would be best for this concrete task, if one is concerned with optimizing energy usage of the networking infrastructure alone. However, if the actual power usage of utilized computing nodes is also included along with the switch's power consumption, then the results suggest that using 2–4 nodes would be optimal for this specific task. Although, in this specific example, we show results, based on a transmission rate of 250 Mbps; other transmission rates can be estimated and stored as knowledge for the PBC.

4.2 Data-based estimate

Using data based estimation, the PBC can calculate the energy usage of a single port based on the available energy profile for a particular switch at a particular transmission speed (throughput). The estimation of the power usage of the whole switch is obtained by adding the baseline switch energy usage and the power consumed by the number of active switch ports. This method allows us to extrapolate the energy consumption of multiple switches of the same or different types by using data obtained from a single device.

For the estimation of the energy required for a single switch port we use formula 2, where the *estimated power per port* is denoted as P_{pp} and the *average power for a specific throughput level*, the *average baseline power*, and the *number of ports* used for the creation of the profile such as P_{tp}, P_b, and N_p respectively.

$$P_{pp} = \frac{P_{tp} - P_b}{N_p} \tag{2}$$

The total switch energy usage P_{total} is given by the *number of used switch ports* multiplied by P_{pp}. The total energy usage for a given task (P_{task}) is the product of P_{total} and the *time* T, which is the time distribution of a task or the total transfer time for a given amount of data.

$$P_{task} = P_{total} * T \tag{3}$$

The data based estimation involves the following parameters of PBC: the number of nodes, total amount of data

---

[1] The current prototype *PBC* implementation is available at https://github.com/zupper/cluster-efficiency.

Fig. 8 Total switch (es) power consumed for task (250 Mbit)

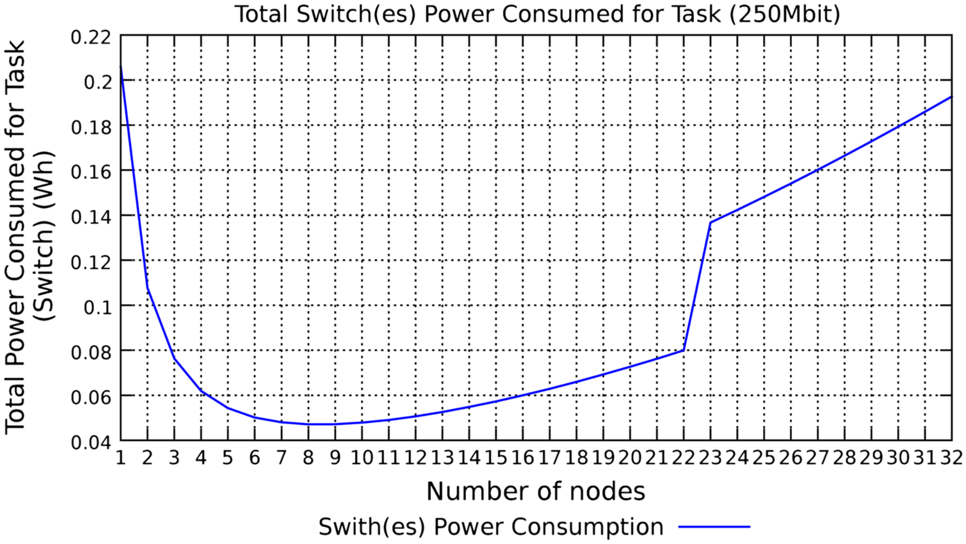
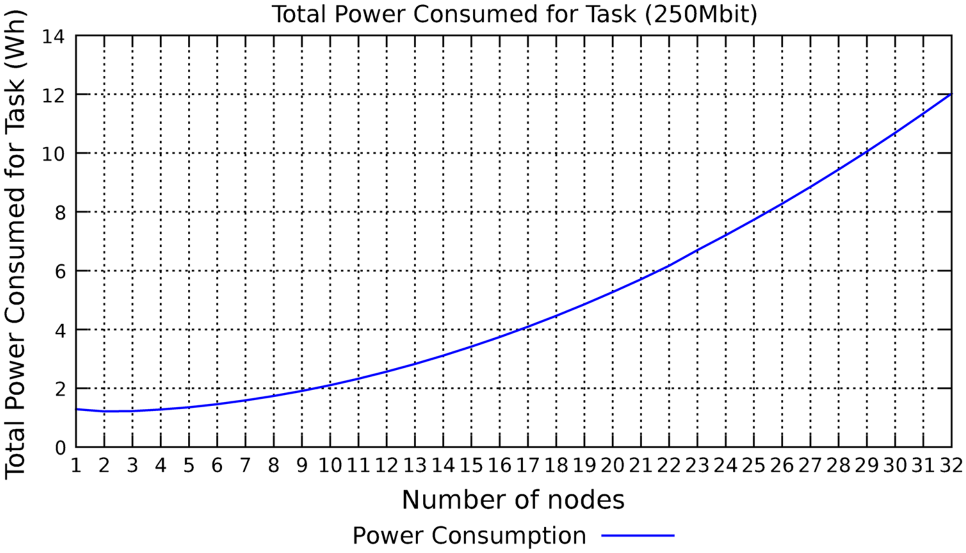


Fig. 9 Total power consumed for task (250 Mbit)



to transfer, the switch's profile, number of usable ports per switch, and total time in minutes to complete the task. By analyzing the existing power profile of the switch, the PBC can estimate the power usage of switches at different transmission rates and find the optimal transmission rate to achieve the lowest power consumption.

The output of the bandwidth estimation tool is given in Fig. 10 and 11. This result was achieved by using the input values shown in Table 2.

We can see a switch needs less time to transfer the fixed amount of data when it uses higher transmission rates, while the power consumption increases linearly before reaching the maximum. We can see an overall hyperbolic decrease in the consumption of power as the transmission rate increases. This means that it will always be more energy-efficient to transmit at faster rates when the task time is flexible and dependent on the time needed for the data transfer. Note how the plot does not include speeds lower than 250 Mbps, as at such low speeds the transfer of the given data for the maximum amount of time would not be possible.

Using the PBC, applications such as power management systems or users such as system administrators can estimate the overall energy consumption of running a given task. This information can help task scheduling services to allocate resources, and set up execution configurations to improve the energy efficiency of a cluster. In the next section, we will discuss how these calculators interact with an energy information monitoring system for effective power management.

Table 1 Parameters used for the task-based estimation tool

Parameter	Description	Values
Nodes	The number of available nodes that *could* be used for this task	2
Switch	The switch's profile to use for energy consumption values	Huawei
Ports/switch	Usable ports per switch	23
Total time	The total time in minutes needed to complete the task on one node	1
Communication cost	Communication cost to exchange messages between nodes	0.05
Speeds	The list of network transmission speeds to use	250
Node power	The average power a node uses while computing	65

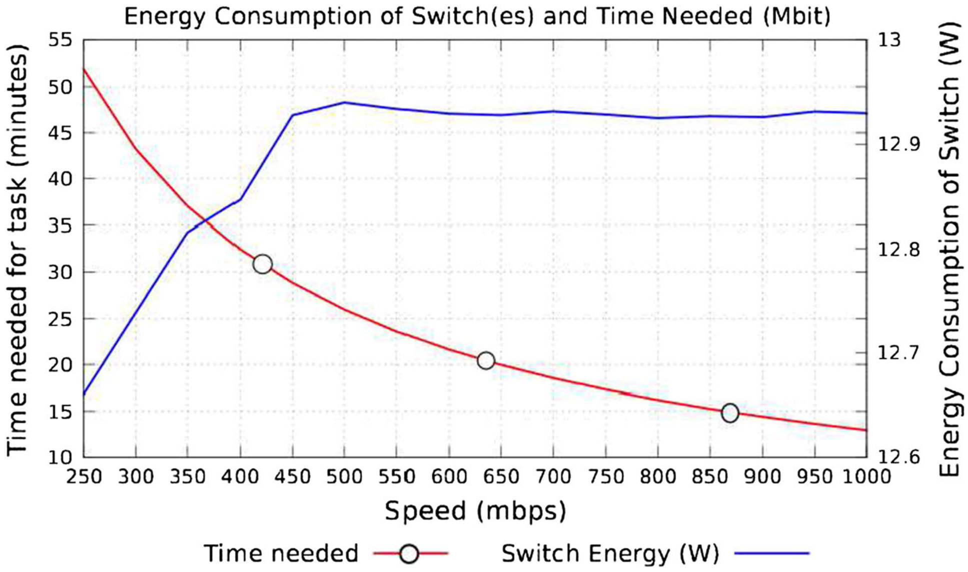


Fig. 10 Power distribution versus time needed for transfer

5 Power Budget Calculator in distributed infrastructure

To serve computing management systems for energy-aware task scheduling or resource allocation, *Power Budget Calculator* requires power profiling information and component information of infrastructures. In this section, we discuss how the Power Budget Calculator can be enhanced by a service called the Energy Knowledge Base (EKB) [19], which was proposed in our group.

5.1 The Energy Knowledge Base and energy description language

The Energy Knowledge Base system is a semantic information system which allows users to obtain information about energy use within an infrastructure using semantic queries, such as energy consumptions at a specific time point, and the expected level of energy consumption during a certain period of time. The information model of EKB is called the

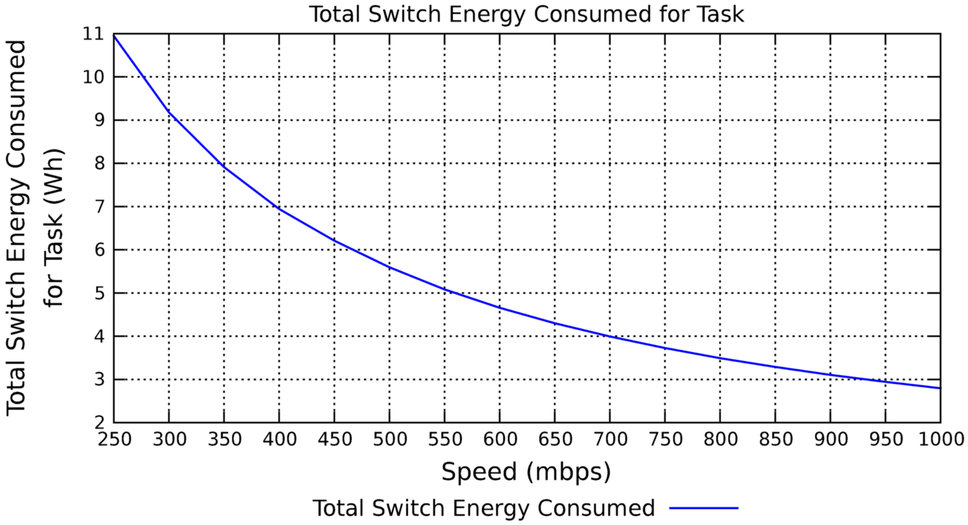


Fig. 11 Total power consumption as a function of the transmission rate

Table 2 Parameters used for the task-based estimation tool

Parameter	Description	Values
Nodes	The fixed number of nodes used for this task	2
Data (MB)	The fixed amount of data that needs to be transferred during the execution of this task	97280
Switch	Which switch's profile to use for energy consumption values	Huawei
Ports per switch	Usable ports per switch	23
Total time	The total time in minutes available for the execution of this task	55

Energy Description Language (EDL)[2] which includes a common vocabulary for different infrastructure domains and a set of energy-related attributes of resources present in infrastructures. EDL contains three main classes: *EnergyMetric*, *MonitorComponent* and *ResourceEnergyDescription*, as shown in Fig. 12.

The *EnergyMetric* class defines two types of energy metrics: the *ObservedMetric* and the *CalculatedMetric*. EKB obtains the value of the former in *ObservedMetric* directly from Power Distribute Units (PDU) installed in infrastructures, while the value of a *CalculatedMetric* is based on numerical calculations on the value of an *ObservedMetric* and of a *PerformanceMetric*. The *ActiveEnergy* and the *PowerAvg* represent the energy consumption and the average real power. We also define the *PowerCapping* to represent the maximum power resource consumes. The *PowerEfficiency* is a measure of the rate of computation or transmission that can be processed by a computer for every Watt of power consumed. Comparably, the *EnergyEfficiency* measures the number of operations or the bytes of data transmission for every joule of energy consumed. Besides the above metrics, EDL allows the extension of metrics.

The *MonitorComponent* class describes the devices that can measure the energy value, such as PDU and other power meters. Subclasses in the *ResourceEnergyDesc* represent information of non-measurable attributes, such as the type of energy source, power state and low-power capabilities.

5.2 The integration of Power Budget Calculator

Energy Knowledge Base system complements the *Power Budget Calculator* with necessary information through an interface at runtime. Tables 1 and 2 illustrate the input

---

[2] The EDL ontology is available at: https://bitbucket.org/hzhu/edl

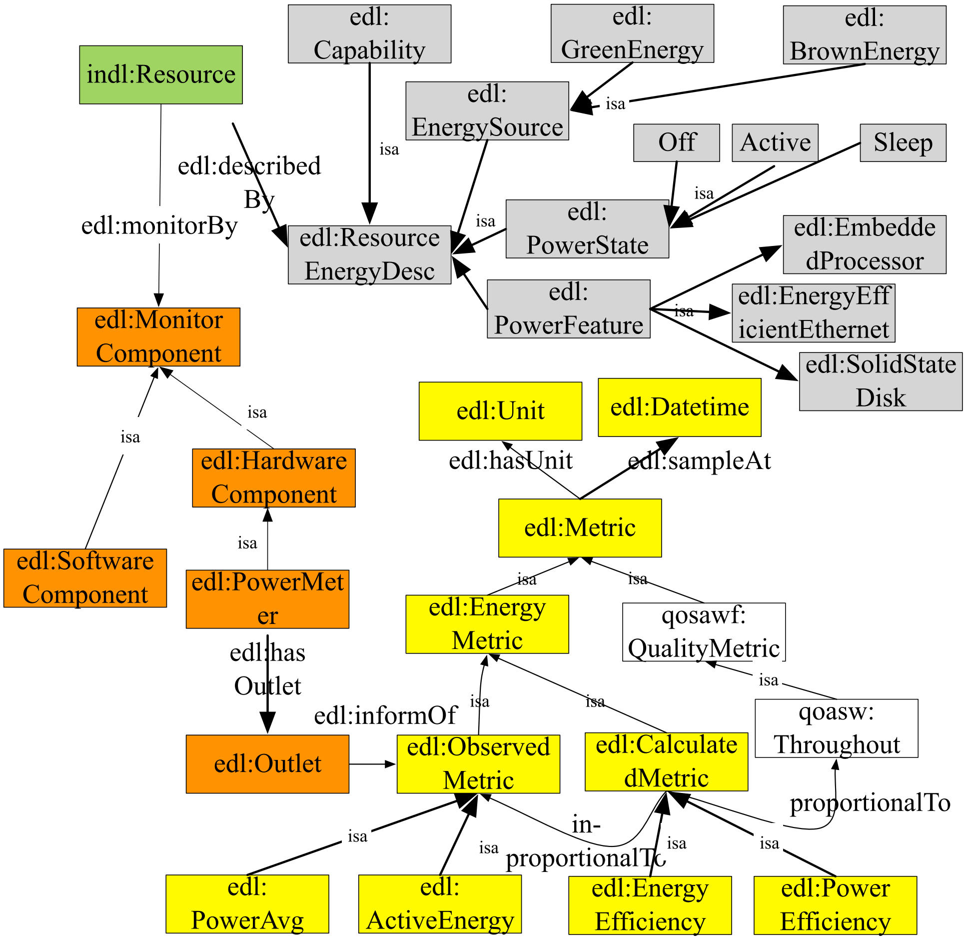


Fig. 12 Energy description language-EDL and its three main parts: the Energy metric class and its subclasses the monitor component class and its subclasses and the resource energy description and its subclasses

information that the PBC consumes. The power profiling of a 802.3az compliant device has been created and stored in EKB as knowledge during the period of experiments; the component information such as devices information and network topologies are instantiated in EKB using another infrastructure ontology called INDL(Infrastructure Description Language) - developed by Ghijsen et al. [20] and which was imported by EDL.

Figure 13 shows an example that PBC obtains the power information from EKB. PBC in the application layer sends a query to EKB. EKB contains a *Query Manager Agent*, an *Energy Information Repository* and a *Resource Collector and Descriptor Agent*. The *Query Manager Agent* receives the request from PBC and translates it into a semantic SPARQL [21] query. After that, this query is imposed to *Energy Information Repository*. Meanwhile the *Query Manager Agent* is responsible for returning results. The *Resource Collector and Descriptor Agent* work in backend to collect information of infrastructures from various sources: PDUs, devices using the SNMP protocol or an existing information service. The obtained information is instantiated using the EDL ontology and stored in the *Energy Information Repository* as RDF triples [22].

The information flow indicated in the Fig. 13 demonstrates the interaction between PBC and EKB, and between EKB and infrastructures. EKB improves the efficiency of PBC. The *Query Manager Agent* provides interfaces of web services. It is thus feasible for EKB to enhance the Power Budget Calculator to obtain real time energy information from an infrastructure to estimate energy consumption of a specified task or device. Our experiment shows the average response time of one query from a store with one million triples is less than 500ms.

EKB improves the data quality of PBC. The current prototype of EKB uses Sesame [23] to implement the *Energy Information Repository*. EDL is developed using OWL [24]. The *Query Manager Agent* supports SPARQL. Data in EKB is in RDF format, which machines can understand, so that EKB supports ontology-based information retrieval instead of key-word based information retrieval to provide more proper information for PBC.

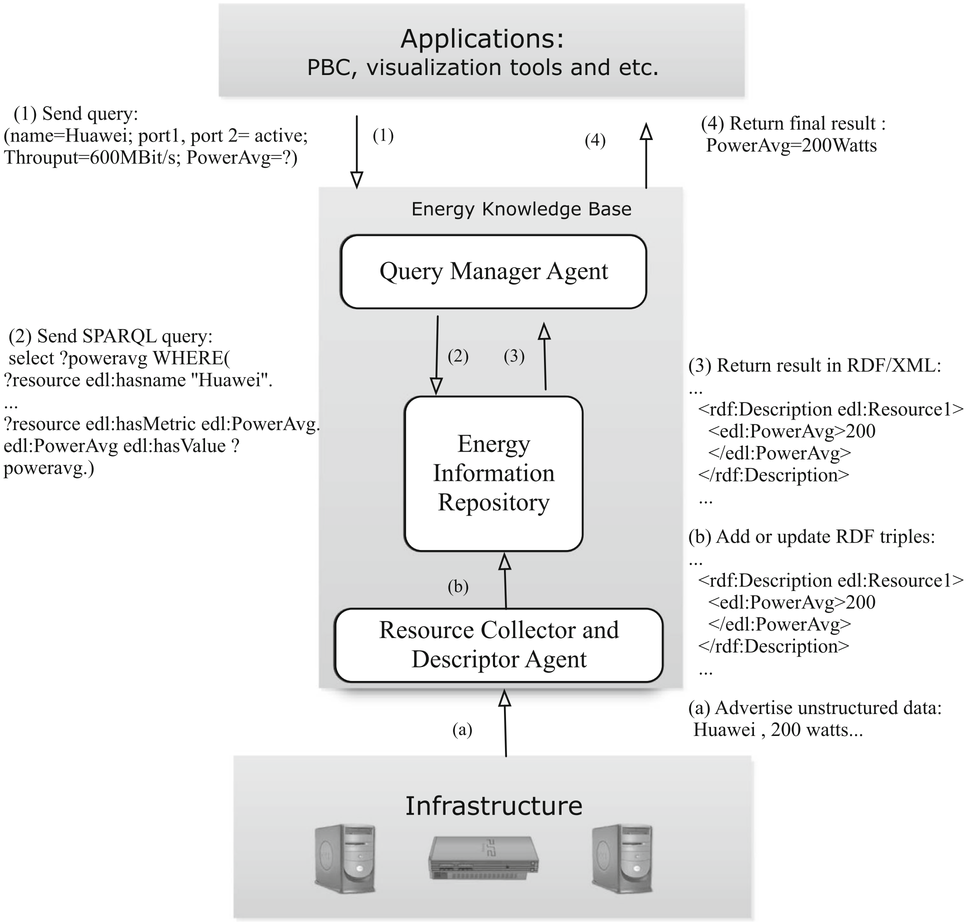


Fig. 13 Power Budget Calculator obtains power profiling from EKB

6 Related work

Extensive research has been done on energy saving in networks.

We start with an overview of existing work on EEE and power models for Ethernet switches (Sect. 6.1) and then we compare research on Power Save Mode (PSM) in wireless networks with our work on EEE (Sect. 6.2). Then we follow with a review of non-sleep techniques used to create energy efficient networks (Sect. 6.3), and we conclude with comparable power calculators (Sect. 6.4).

6.1 EEE research

Reviriego and his group [25,26] have tested the energy consumption of EEE switches under different link utilization with various numbers of active ports. They proposed a simple power model to predict the energy savings when deploying new switches. A major difference with our work is that we looked at the energy consumption of switches given different frame arrival intervals. This showed that the energy saving depends on the duration of sleep time and number of state transitions. Moreover, our evaluations clearly indicated that the energy characteristics of switches varies by vendor implementation. The power model we presented is based on our experiments, and it can be easily updated to incorporate knowledge about other switches.

There has been research done in defining power models of non-EEE Ethernet devices. Mahadevan et al. [27] built such a model to account for the energy consumption of non-EEE network devices. The model states that the power consumption of a switch only depends on the chassis, the number of line cards, the number of ports and the speed of links. This study showed that utilization of each port, packet sizes and ternary content addressable memory (TCAM) hardly influenced the power consumption. Such a model cannot be considered in our case given we know that for EEE switches power consumption depends on the throughput.

Wang's model [28] is not built for EEE devices, but the parameters in it are similar to ours, as it includes the power consumed by the chassis, the line cards and the ports, and the power consumption of the ports depends on the data rate of outgoing and incoming flows.

[29] presented an analytical power model by predicting the time spent in each state of the whole EEE transmission cycle. This model can be used to estimate energy savings, but it has the drawback that it is accurate only in the specific environment of the considered use case. [30] evaluated the potential power saving of EEE on 1Gbps link using real traffic traces and compared the result with the analytical model from [29]. Our model is usable in more generic environments.

The EEE protocol leaves the opportunity to implement different power saving policies based on sleep and wake transitions. The simple and common approach is to wake up a link from quiet mode when a new frame arrives and it needs to deliver it. This behaviour could cause oscillation between quiet and active, which is not energy efficient. Recent research has focused on burst transmission or called packet coalescing to enhance power savings for EEE switches [16,31–37].

An observation is that energy consumption can be greatly reduced if the link does not wake up for transmission until a large number of frames arrive. To minimize the transmission delay the threshold for number of frames buffered before waking up should be carefully controlled. [31] proposed an analytical model of energy saving and delay time, which can be used to find out the optimal trade-off between the energy saving and delay. Maestro et al. [37] also studied the combination of EEE and adaptive link rate. Interfaces can adaptively make the decision of switching into sleep or just scaling link rate according to the current traffic.

Besides burst transmission, other optimizations of EEE have been analyzed. [38] uses a coordinated transmissions scheme. This scheme staggers the time of sending and receiving traffic to make sure only one link direction is active at any time. The benefit is to further reduce the power usage in low power mode because the circuitry related to unwanted signals can also go to sleep.

6.2 Power saving model in wireless networks

Similar to EEE, PSM is a sleep mechanism. It allows 802.11 wireless interfaces to switch to a lower power mode. In this mode, all interfaces in the network are synchronized to wake up periodically (this period is called the beacon interval) to listen to messages. The access point buffers the messages for the interfaces, and broadcasts during the period when they all are awake. Anastasi et al. [39,40] analyzed the PSM energy consumption with respect to average burst size, transport-level throughput and the number of users when downloading fixed-sized files. The analysis showed PSM was effective during traffic bursts. But PSM was not optimal in the case of long inactivity periods between bursts. Thus, the authors proposed an energy manager to dynamically tune between enabling PSM and powering off. This method was not included in the 802.11 protocol. Tauber et al. [41] did some experiments on the real testbed and investigated the impact of enabling/disabling PSM on energy and performance under different traffic characteristics such as data rate and packet size. They concluded that no significant benefit could be gained using PSM in their scenarios. This conclusion is different from 802.3az in our case.

6.3 Other technologies for green Ethernet

Link sleep we describe above is only one of the technologies for the green Ethernet. *Adaptive Link Rate* and *Energy-aware Routing* are also widely used to conserve energy in the data transmission.

6.3.1 Adaptive link rate

In general, operating an Ethernet communication link at a lower rate can enable energy reduction [42]. Some studies [43,44] have been conducted on the adaption of link speed for achieving a tradeoff between performance and energy consumption. However, these technologies face high overhead due to the required auto-negotiation of link speeds between the endpoints. [45] used big buffering and path diversity to mitigate performance impact during link rate switch. But it is difficult to predict traffic pattern at runtime in data centers. Gunaratne et al. [46,47] proposed a two-way MAC frame handshake mechanism for fast data rate switching. But this mechanism is not mature enough to be implemented. They designed a dual threshold and link utilization based policy to scale link rate. This cuts down the frequency of rate switches and maximizes the energy savings. Wang et al. [28] analyzed the correlation between flows, and consolidated the selected flows to power off unused switches while meeting individual flow's peak demand.

6.3.2 Energy-aware routing

Energy-aware routing reduces energy usage by aggregating the traffic over a subset of network links or network devices in over-provisioned networks. The unused network links or devices can be powered off. In [48], the authors focused on energy efficiency in access networks: they achieved energy reduction by aggregating user traffic in fewer gateways and by scheduling active links between line cards to power off the line cards when possible. Chabarek et al. [49] focused on how to save energy of aggregate traffic through optimal route selection and configuration in wide-area networks. Honeyguide [50] optimized the energy consumption of VM replacement on the network. They extended tree-based topologies to bypass links and turn off unused switches without contradicting with original replica replacement policy.

Recently, more attention has been paid to data center networks because of their dominant contribution to energy consumption. Heller et al. [51] presented a network-wide energy optimizer for fat-tree data centers called ElasticTree. ElasticTree generates a subset of fixed topologies to keep network elements in the selected topology active and turns off as many unneeded links and switches as possible. The authors of [52,53] proposed a new energy efficient approach to control traffic that aims to solve the inevitable packet loss present in routing optimization technologies. They aggregated the input links before entering in the switches or routers instead of routing traffic. Li et al. [54] enabled the second backup Ethernet port to build a cost effective data center network structure that requires fewer switches. As a survey, [55] compared the energy characteristics under different data center network architectures. They found the energy consumption gap was quite small if employing existing green network technologies like sleeping and path selection from over-provisioned networks.

6.4 Power calculator tools and information monitor system

In Sect. 4 we presented our power calculator tool which included the model for the energy characteristics of 802.3az compliant devices and that was suitable to predict the energy consumption for applications operating in cluster environments. A couple of other calculators have been proposed, and we compare them to ours.

Oracle's Sun Power calculator [56] helped users understand the overall power consumption of servers and storage systems according to the configuration on hardware. It is very coarse-grained because it only allows users to specify the percentage of hardware utilization. Cisco [57] provides an online tool to calculate power consumption of networking gear given the model and configuration information. They can calculate the dynamic power usage such as the idle power and 50 % load power of switches when specifying information at runtime, such as the number of the port used. However, all the calculators above are not workload-aware, and cannot be directly used to measure energy consumption of different workloads. Our calculator can account for the energy consumption of different application scenarios according to the information from workload, such as the amount of transferred files. Using our calculator, the system administrator can determine an optimal network device configuration for applications in a cluster environment.

Our calculator depends on data and metadata of infrastructures. Several systems for information monitoring are similar to EKB. Globus has developed the monitoring and discovery system (MDS) [58] which is able to aggregate information about resources and federate with related monitors. PerfSONAR [59] is an infrastructure for network performance monitoring, which can respond to queries of performance measurements and discover services registered in a federated environment. In comparison, EKB is a semantic system for energy monitoring, which can expose complete and well-organized energy information to users for power management.

7 Conclusions

We have discussed several experiments analyzing the 802.3az protocol to determine achievable energy savings of switches. A result of these experiments is our Power Budget Calculator prototype, which allows users to account for the network energy savings when planning task distributions in HPC environments.

We determine that energy savings are possible with the vast majority of traffic patterns. The current implementations of 802.3az are effective for energy savings of low utilization links, but are limited when it comes to intensive computing. We believe it is possible to save even more power when taking an active optimization strategy in the transport layer or link layer.

We implement a Power Budget Calculator prototype to model the power consumption of 802.3az-enabled networks when executing distributed data intensive applications. It can estimate the execution conditions under which specific applications will use the least amount of power and can be incorporated into the operation of cluster or grid scheduling and workflow systems.

We integrate PBC into our Energy Knowledge Base for seamless power profiling access. With the help of EKB, PBC can better estimate the energy consumption of distributing the tasks onto different computer resources.

8 Future work

We will investigate the effect of network optimizations without modifying the protocol in future work. For example, we will study the influence of flow control technologies like Nagle's algorithm in the transport layer.

There is more work we intend to perform to improve the usability of our PBC calculator. We will make the assumptions more close to reality, for instance, multiple tasks, heterogeneous network devices and complicated network architecture should be covered.

Moreover, EKB faces the problem of scalability, and the calculator cannot work across different network domains. We will improve the implementation of EKB for better support. In the future, the budget calculator will be coupled with the workflow planner [60] developed in our earlier projects to provide energy efficient workflow planning and scheduling in network environment.

Acknowledgments We would like to thank the FP7 EU funded Integrated project ENVRI project (project number 283465) and the Dutch national and education network SURFnet, through the Giga-Port Research on Network (RoN) project, the Dutch national program COMMIT and the NWO-funded project GreenClouds for sponsoring this research. We would also like to thank Cisco and Huawei for lending us their switches. Thanks Dr. Paul Martin from University of Edinburgh for proof reading.

References

1. Gokhale, M., Cohen, J., Yoo, A., Miller, W., Jacob, A., Ulmer, C., Pearce, R.: Hardware technologies for high-performance data-intensive computing. IEEE Comput. **41**(4), 60–68 (2008)
2. D-link, Green Ethernet. http://dlinkgreen.com/energyefficiency.asp. Accessed July 2014
3. IEEE Standard Associate, IEEE Std 802.3az-2010. http://standards.ieee.org/findstds/standard/802.3az-2010.html. Accessed July 2014
4. Pavlov, D., Soert, J., Grosso, P., Zhao, Z., van der Veldt, K., Zhu, H., de Laat, C.: Towards energy efficient data intensive computing using IEEE 802.3az. In: The 2012 International Workshop on Data-Intensive Scalable Computing Systems (DISCS) in Conjunction with the 2012 ACM/IEEE Supercomputing Conference (SC) (2012)
5. Gupta, M., Singh, S.: Greening of the Internet. Anchor Books, New York (2003)
6. Gupta, M., Grover, S., Singh, S.: A feasibility study for power management in LAN switches. In: Proceedings of the 12th IEEE International Conference on Network Protocols, pp. 361–371 (2004)
7. Gupta, M., Singh, S.: Using low-power modes for energy conservation in Ethernet LANs. In: IEEE INFOCOM - 26th IEEE International Conference on Computer Communications, pp. 2451–2455 (2007)
8. Herreria-Alonso, S., Rodriguez-Perez, M., Fernandez-Veiga, M., Lopez-Garcia, C.: Opportunistic power saving algorithms for Ethernet devices. Comput. Netw. **55**(9), 2051–2064 (2011)
9. Ananthanarayanan, G., Katz, R.H.: Greening the Switch. In: Proceedings of the Conference on Power Aware Computing and Systems (HotPower) (2008)
10. Reviriego, P., Hernadez, J., Larrabeiti, D., Maestro, J.A.: Burst transmission in Energy Efficient Ethernet. IEEE Internet Comput. **14**(4), 50–57 (2010)
11. Cisco Systems, Inc., Cisco SG300-28 Switch. http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps10898/brochure_c02-610054.pdf. Accessed Dec 2012
12. Huawei S1728GWR-4P Switch. http://market.huawei.com/hwgg/enterprise/u-channel/pdf/S1700.pdf. Retrieved Dec 2012
13. Intel I350–T2 Nic. http://intelethernet-dell.com/wp-content/uploads/2012/04/Dell_Intel-Ethernet-I350-T2_T4-PB_adkit.pdf. Retrieved Aug 2012
14. Reviriego, P., Hernandez, J., Larrabeiti, D., Maestro, J.A.: Performance evaluation of energy efficient ethernet. IEEE Commun. Lett. **13**(9), 697–699 (2009)
15. Mausezahn Website. http://www.perihel.at/sec/mz/. Retrieved Dec 2012
16. Christensen, K., Florida, S., Nordman, B., Bennett, M., Berkeley, L.: IEEE 802. 3az : the road to Energy Efficient Ethernet. IEEE Commun. Mag. **48**(11), 50–56 (2010)
17. Nagle, J.: Congestion control in IP/TCP internetworks. SIGCOMM Comput. Commun. Rev. **14**(4), 11–17 (1984)
18. Chen, Q., Grosso, P., van der Veldt, K., de Laat, C., Hofman, R., Bal, H.E.: Profiling energy consumption of vms for green cloud computing. In: Proceedings of the International Conference on Cloud and Green Computing (CGC) (2011)
19. Zhu, H., van der Veldt, K., Grosso, P., Zhao, Z., Liao, X., de Laat, C.: Energy-aware semantic modeling for large scale infrastructures. In: Proceedings of the IEEE International Conference on Green Computing and Communications (GreenCom) Work In Progress Session, pp. 11–14 (2012)
20. Ghijsen, M., van der Ham, J., Grosso, P., de Laat, C.: Towards an Infrastructure Description Language for Modeling Computing Infrastructures. In: 10th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), pp. 207–214. Madrid (2012)
21. Prudhommeaux, E., Seaborne, A.: SPARQL query language for RDF. W3C Recomm. **4**, 1–106 (2008)
22. Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, D., Broekstra, J.: The semantic Web: the roles of XML and RDF. IEEE Internet Comput. **15**(3), 63–74 (2000)
23. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: a generic architecture for storing and querying RDF and RDF Schema. In: Horrocks, I., Hendler, J. (eds.) Proceedings of the first Int'l Semantic Web Conference (ISWC 2002). Lecture Notes in Computer Science, vol. 2342, pp. 54–68. Springer, Sardinia (2002)
24. Dean, M., Schreiber, G.: OWL Web Ontology Language Reference. W3C Recommendation, W3C, New York (2004)
25. Reviriego, P., Christensen, K., Rabanillo, J., Maestro, J.A.: An initial evaluation of Energy Efficient Ethernet. IEEE Commun. Lett. **15**(5), 578–580 (2011)
26. Reviriego, P., Sivaraman, V., Zhao, Z., Maestro, J.A., Vishwanath, A., Sanchez-Macian, A., Russell, C.: An energy consumption model for Energy Efficient Ethernet switches. In: International Conference on High Performance Computing & Simulation (HPCS), pp. 98–104 (2012)
27. Mahadevan, P., Sharma, P., Banerjee, S.: A power benchmarking framework for network devices. In: Proceedings of the 8th International IFIP-TC 6 Networking Conference, pp. 795–808 (2009)
28. Wang, X., Yao, Y., Wang, X., Lu, K., Cao, Q.: CARPO: correlation-aware power optimization in data center networks. In: Proceedings IEEE INFOCOM, pp. 1125–1133 (2012)
29. Marsan, M.A., Anta, A.F., Mancuso, V., Rengarajan, B., Vasallo, P.R., Rizzo, G.: A simple analytical model for Energy Efficient Ethernet. IEEE Commun. Lett. **15**(7), 773–775 (2011)
30. Mancuso, V., Chatzipapas, A.: On IEEE 802.3az energy efficiency in Web Hosting Centers. IEEE Commun. Lett. **16**(11), 1880–1883 (2012)
31. Herreria-Alonso, S., Rodriguez-Perez, M., Fernandez-Veiga, M., Lopez-Garcia, C.: Optimal configuration of Energy-Efficient Ethernet. Comput. Netw. **56**(10), 2456–2467 (2012)
32. Herreria-Alonso, S., Rodriguez-Perez, M., Fernandez-Veiga, M., Lopez-Garcia, C.: A power saving model for burst transmission in Energy-Efficient Ethernet. IEEE Commun. Lett. **15**(5), 584–586 (2011)
33. Herreria-Alonso, S., Rodriguez-Perez, M., Fernandez-Veiga, M., Lopez-Garcia, C.: How efficient is Energy-Efficient Ethernet? In: The 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp. 1–7 (2011)
34. Maestro, J.A.: Optimize Energy Efficient Ethernet performance in bundled links. EDN network pp. 23–29 (2011)
35. Mostowfi, M., Christensen, K.: Saving energy in LAN switches: new methods of packet coalescing for Energy Efficient Ethernet. (2011). doi:10.1109/IGCC.2011.6008547
36. de la Oliva, A., Hernandez, T.R., Guerri, J.C., Hernandez, J.A., Reviriego, P.: Performance analysis of energy efficient ethernet on video streaming servers. Comput. Netw. **57**(3), 599–608 (2013)
37. Reviriego, P., Maestro, J.A., Hernandez, J., Larrabeiti, D.: Study of the potential energy savings in Ethernet by combining Energy

Efficient Ethernet and adaptive. Trans. Emerg. Telecommun. Technol. **23**(3), 227–233 (2012)
38. Reviriego, P., Christensen, K., Maestro, J.A.: Using coordinated transmission with Energy Efficient Ethernet. In: Proceedings of the 10th International IFIP TC 6 Conference on Networking- Volume Part I, pp. 160–171 (2011)
39. Anastasi, G., Conti, M., Gregori, E., Passarella, A.: Saving energy in Wi-Fi hotspots through 802. 11 PSM : an analytical model. In: Proceedings of the Workshop on Linguistic Theory and Grammar Implementation, pp. 24–26 (2004)
40. Anastasi, G., Conti, M., Gregori, E., Passarella, A.: 802.11 power-saving mode for mobile computing in Wi-Fi hotspots: limitations, enhancements and open issues. Wirel. Netw. **14**(6), 745–768 (2007)
41. Tauber, M., Bhatti, S.N.: The effect of the 802.11 power save mechanism (PSM) on energy efficiency and performance during system activity. In: Proceedings of the IEEE International Conference on Green Computing and Communications (GreenCom) (2012)
42. Gunaratne, C., Christensen, K., Nordman, B.: Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed. Int. J. Netw. Manag. **15**(5), 297–310 (2005)
43. Mahadevan, P., Sharma, P., Banerjee, S., Ranganathan, P.: Energy aware network operations. In: IEEE INFOCOM Workshops, pp. 1–6 (2009)
44. Nedevschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., Wetherall, D.: Reducing network energy consumption via sleeping and rate-adaptation. In: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, pp. 323–336 (2008)
45. Abts, D., Marty, M.R., Wells, P.M., Klausler, P., Liu, H.: Energy proportional datacenter networks. In: Proceedings of the 37th annual international symposium on Computer architecture(ISCA), p. 338. New York, (2010)
46. Gunaratne, C., Christensen, K., Nordman, B., Suen, S.: Reducing the energy consumption of ethernet with adaptive link rate (ALR). IEEE Trans. Comput. **57**(4), 448–461 (2008)
47. Gunaratne, C., Christensen, K., Suen, S.W.: Ethernet adaptive link rate (ALR): analysis of a buffer threshold policy. In: IEEE Globecom 2006, pp. 1–6. Ieee (2006)
48. Goma, E., Canini, M., Toledo, A.L., Laoutaris, N., Kosti, D.: Insomnia in the access or How to curb access network related energy consumption. In: Proceedings of the ACM SIGCOMM Conference, pp. 338–349 (2011)
49. Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang, D., Wright, S.: Power awareness in network design and routing. In: IEEE INFOCOM-The 27th Conference on Computer Communications, pp. 457–465 (2008)
50. Shirayanagi, H., Yamada, H., Kono, K.: Honeyguide: A VM migration-aware network topology for saving energy consumption in data center networks. In: 2012 IEEE Symposium on Computers and Communications (ISCC), pp. 460–467 (2012)
51. Heller, B., Mahadevan, P.: ElasticTree : saving energy in data center networks. In: The 7th USENIX Conference on Network Systems Design and Implementation(NSDI), pp. 2–17 (2010)
52. Singh, S., Candy, Y.: Putting the cart before the horse: merging traffic for energy conservation. IEEE Commun. Mag. **49**(June), 78–82 (2011)
53. Yiu, C., Singh, S.: Merging traffic to save energy in the enterprise. In: Proceedings of the 2nd International Conference on Energy-Efficient Computing and Networking (e-Energy), p. 97. New York, (2011)
54. Li, D., Guo, C., Wu, H., Tan, K., Zhang, Y., Lu, S., Wu, J.: Scalable and cost-effective interconnection of data-center servers using dual server ports. IEEE/ACM Trans. Netw. **19**(1), 102–114 (2011)
55. Shang, Y., Li, D., Xu, M.: A comparison study of energy proportionality of data center network architectures. In: 32nd International Conference on Distributed Computing Systems Workshops, pp. 1–7 (2012)
56. Oracle. Sun power calculators. http://www.oracle.com/us/products/servers-storage/sun-power-calculators/index.html (2012). Retrieved Aug 2012
57. Cisco UCS Power Calculator. http://www.cisco.com/assets/cdc_content_elements/flash/dataCenter/cisco_ucs_power_calculator/ (2012). Retrieved Aug 2012
58. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid information services for distributed resource sharing. In: Proceedings. 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–194 (2001)
59. Hanemann, A., Boote, J.W., Boyd, E., Dur, J., Kudarimoti, L., Lapacz, R., Swany, D.M., Trocha, S., Zurawski, J.: Perfsonar: a service oriented architecture for multi-domain network monitoring. In: In Proceedings of the Third International Conference on Service Oriented Computing (ICSOC 2005) (2005)
60. Zhao, Z., Grosso, P., van der Ham, J., Koning, R., de Laat, C.: An agent based network resource planner for workflow applications. Int. J. Multiagent Grid Syst. **7**(6), 187–202 (2011)

Hao Zhu is a PhD student in the SNE group at the UvA. His research interests are in the area of Green IT, cloud computing and semantic web. To be particular, he studies power estimation models, power management of distributed system and semantic descriptions of energy-aware infrastructures.

Karel van der Veldt is a PhD candidate in the SNE group of the University of Amsterdam. He obtained his master's in Computer Science at the University of Amsterdam and his bachelor in Computer Engineering at the Hogeschool van Amsterdam. His research interests are green computing in networks and data centers, and high performance and distributed computing.

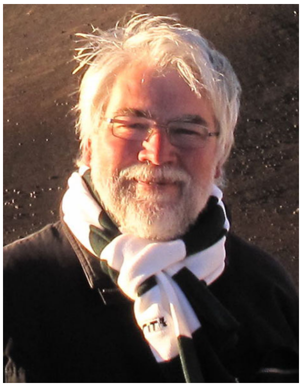
Zhiming Zhao obtained his PhD in computer science in 2004 from University of Amsterdam (UvA). He is currently a senior researcher in the System and Network Engineering group at UvA. He leads the scientific linking model research in the FP7 ENVRI project, and developed a Network QoS aware workflow planning system in the context of CineGrid, Gigaport and Geyses project. His research interests include advanced network, workflow management systems, multi agent system and big data research infrastructures. http://www.science.uva.nl/~zhiming.

Paola Grosso is assistant professor in the SNE group at the UvA. She is the lead researcher of the group activities in the field of GreenIT, optical networking and distributed infrastructure information modelling. She participates in several national and international projects, such as COMMIT, GreenClouds, Green Software and MOTE. Her research interests are green ICT, provisioning and design of hybrid networks for lambda services; development of information models for hybrid multi-domain multilayer networks. http://www.science.uva.nl/~grosso.

Dimitar Pavlov studied master program on computer networking at University of Amsterdam between 2011–2012. He is current a co-Founder of Black Bear Solutions BV. His main research interests are energy efficient ethernet, distributed systems and mobile computing.

Joris Soeurt studied master program on computer networking at University of Amsterdam between 2011–2012. He is currently a core network engineer at Tele2 Netherlands. His main research interests are energy efficient ethernet, network security, and 4G network.

Xiangke Liao received his BS degree in the Department of Computer Science and Technology from Tsinghua University, Beijing, China, in 1985, and his MS degree in 1988 from National University of Defense Technology, Changsha, China. He is currently a full professor and the dean of School of Computer, National University of Defense Technology. His research interests include parallel and distributed computing, high-performance computer systems, operating systems, cloud computing and networked embedded systems. He is a member of the IEEE and ACM.



Cees de Laat is chair of the System and Network Engineering research section at the University of Amsterdam. Research covers optical and switched networking and workflows for processing of big data in PetaScale e-Science applications, Semantic Web to describe e-infrastructure resources, information complexity, Authorization architectures and Systems Security & privacy of information in distributed environments. He serves as gfsg member of Open Grid Forum, is chair of GridForum.nl and serves on the Lawrence Berkeley Laboratory Policy Board on matters regarding ESnet, is co-founder and organizer of several past meetings of the Global Lambda Integrated Facility (GLIF) and founding member of CineGrid.org. http://delaat.net/.