Measuring and Modeling the Power Consumption of OpenFlow Switches

Fabian Kaup, Sergej Melnikowitsch, David Hausheer
Peer-to-Peer Systems Engineering Lab, Technische Universität Darmstadt
Email: {fkaup|melnikow|hausheer}@ps.tu-darmstadt.de

***Abstract*—The power consumption of network devices contributes to a considerable fraction of the energy expenses of data center and service providers. Recently, Software Defined Networking (SDN) has emerged as a novel networking paradigm that allows optimizing the traffic in a variety of ways, ranging from the Ethernet layer to the network layer and above. This makes SDN a versatile approach for a large number of applications. However, little is known about the power consumption of SDN-enabled networking devices, especially OpenFlow switches. This paper presents measurements and derived power consumption models for two SDN devices, namely an OpenFlow-based hardware switch (NEC PF 5240) and the Open vSwitch running on server grade hardware. The effect of configuration, management, and the managed traffic on the resulting power consumption is evaluated. Based on these measurements, a power model is derived, allowing for an estimation of the power consumption based on the configuration and network traffic only. Being available on the OpenFlow controller, the information about the power model enables an estimation of the power consumption of the full network without additional measurement equipment. The resulting model shows an error of less than 8% for the software switch and less than 1% for the hardware switch.**

I. Introduction

The overall power consumption of Information and Communications Technology (ICT) is estimated to range in between 84 and 143 GW, out of which network devices consume around 7% [1]. This is expected to increase, as the Internet traffic is projected to double within 3-4 years. An increase in energy-efficiency of network devices is possible, as shown by the decreasing cost per GB transferred [2]. Moreover, approaches like switching off components [3], links [4], or reducing the link rate [5], can – given a low load on the network – result in energy savings without affecting the service quality.

The emerging concept of Software Defined Networking (SDN) [6], with its most prominent implementation OpenFlow [7], provides for an increased control over networking devices, independent of vendor specific implementations. SDN enables, e.g., an efficient use of redundant links in datacenter networks [3], or increasing the link utilization to almost 100% [8], which is hardly possible in conventional networks. To fully exploit the potential of SDN in network management on reducing the energy consumption, further information on the underlying hardware is required.

There exist already a number of approaches analyzing the power consumption and improving the energy efficiency of conventional enterprise or consumer grade network switches [9] [10]. However, little is known about the power consumption of OpenFlow switches. Therefore, this paper provides insights into the power consumption of these devices, with the goal to enable energy optimization based on OpenFlow-specific parameters.

The work is led by the three main questions:

- What is the influence of the extended switching functionality provided by OpenFlow on the power consumption of hardware and software switches?
- What are the differences between hardware and software implementations of the OpenFlow feature set?
- How can the energy efficiency of the analyzed devices be improved?

For this, a framework for the generation of OpenFlow power models is proposed, generating power models for all kinds of OpenFlow enabled devices in a simple manner. To validate this approach, power models for a hardware switch (NEC PF 5240) and an Open vSwitch running on a server are presented. The quality of the measured models is assessed using a Floodlight[1] plugin. The plugin compares the measurements of a connected hardware power meter to the estimations generated by the power model and derives the resulting error. Using this plugin, the power consumption of the connected OpenFlow switches is estimated with an accuracy of less than 1% for the hardware switch and 8% for the software switch.

The contributions of this work are three-fold. A power measurement framework is introduced, allowing the simple calibration of OpenFlow power models. A plugin for the Floodlight OpenFlow controller was developed, using the derived power models to estimate the power consumption of connected devices based on the configuration and processed packets. From the differences in the power models, general conclusions for the optimization of the energy-efficiency of hardware and software switches are drawn.

The remainder of this paper is organized as follows. Section II compares different power modeling and energy saving techniques. Section III describes the measurement setup and the framework used to derive the power model. The resulting measurements are described and discussed in Section IV, from which the power models are derived in Section V. Section VI compares the estimated power consumption to the power measurements taken by generating arbitrary traffic on the switches. Finally, Section VII concludes the paper with a discussion of possible applications of the power models.

[1] http://www.projectfloodlight.org/floodlight/ [Accessed: 2014-05-02]

II. Related Work

Most recent publications either focus on the power consumption of conventional network hardware [9]–[12] or the performance of OpenFlow [13], [14].

Hlavacs et al. [9] analyze and model the power consumption of residential and professional switches. They show that for residential switches, the power consumption decreases with a higher number of packets, while the power consumption of enterprise grade switches is constant or slightly increasing. Chabarek et al. [12] analyzed the power consumption of Cisco's 7507 and GSR 12008 and approximate it by a linear model, which depends on the idle consumption, the number of installed line cards, their configuration and the processed traffic rate, concluding that a chassis closely fitting the demand minimizes the network's energy consumption. Mahadevan et al. [10] describe a power measurement framework applicable to conventional network devices. They measure the influence of the number of line cards and active ports, their configured capacity, and utilization. The network device is connected to a traffic generator and receiver, a device configurator, and power meter to configure a specific workload and to measure the resulting power consumption. By monitoring the traffic and power consumption, the analyzer can then derive a power model of the network device. These power models can then be used to optimize routing decisions [11] and network paths [15], [16] in general network environments. The approaches are combined by Bolla et al. [17] into a unified framework allowing energy optimizations over a wide range of devices.

Still, the power consumption of SDN operations is largely unknown. Congdon et al. [13] develop a model to improve the power consumption of OpenFlow switches by including a prediction engine to avoid costly ternary content addressable memory (TCAM) lockups, such reducing the latency by 80% and simultaneously cutting the power demand by almost 90%. Rotsos et al. [14]. focus on the switching performance and configuration delays of hardware and software switches. Three hardware switches of undisclosed type are compared with the Open vSwitch and the NetFPGA. The switching performance differs by a factor of 50 between hardware accelerated and software implemented functions of the hardware switches.

This paper differs from previous studies as it measures the power consumption of individual OpenFlow features applied to individual flows and integrates them into a power model. The presented work extends the methodology proposed by Mahadevan et al. [10] by extending the functionality of a Floodlight controller to estimate the power consumption of the Device under Test (DUT) in real-time

III. Measurement Setup

The measurement setup is chosen similar to [10], but additionally supports measuring OpenFlow related parameters. Derived from this, the test setup consists of the DUT, a traffic generator, a traffic receiver, the power meter, a controller and a test coordinator, as given in Figure 1. To support OpenFlow in this setting, the controller is a modified Floodlight controller, configuring the switch using the OpenFlow protocol.

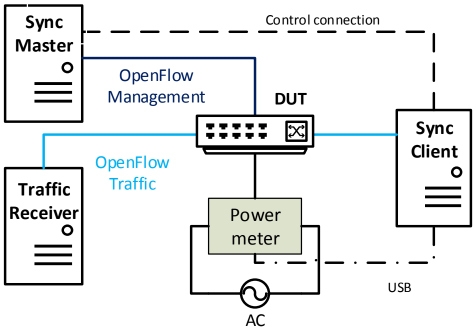


Fig. 1. The measurement setup and wiring

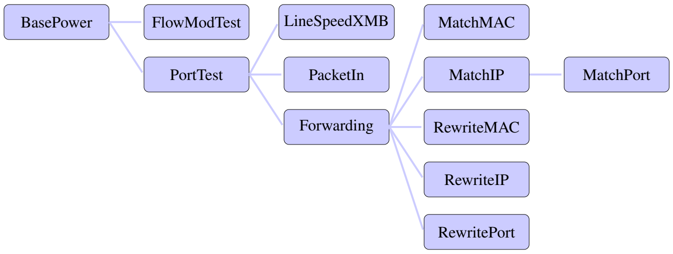


Fig. 2. Dependencies of the different measurement tasks

To test the different OpenFlow commands, a python framework was developed, allowing for a simple extension of the test cases used to stress the device. This is important, considering the quick evolution of the OpenFlow standards. Currently OpenFlow version 1.0 [18] is supported by the presented test suite, which at the time of measurements was the highest version supported by the hardware switch. The different test cases are defined analog to the dependencies of the OpenFlow matches and actions, given in Figure 2. This requires that the influence exerted by one OpenFlow feature on the total power consumption is independent of the underlying OpenFlow features. Hence, the respective measurements are subtracted from each other in order to measure the influence of the particular parameter.

The *SyncMaster* in Figure 1 coordinates the tests and configures the OpenFlow switch and the *SyncClient* for each test, while the *SyncClient* executes the electrical measurements and generates the traffic. The measurements are collected locally and only after the test is over, the data is sent to the *SyncMaster* for aggregation and analysis.

The accuracy of the power measurements of the power meter depends on the measurement range. For a range of 800 W, the resolution is 100 mW. The relative error of 0.8% combined with an error of up to 10 digits leads to an error of 3 W, or 1.13%, for the power consumption of the NEC switch with 264 W idle power. Similarly, the error for the Open vSwitch with a maximum power consumption of 55 W leads to an error of 0.54 W or 0.98%. The sampling interval of the power meter is limited to 0.5 s. Relating the sampling interval to the mains frequency leads to one power sample for every 25 oscillations. This allows the power meter to properly

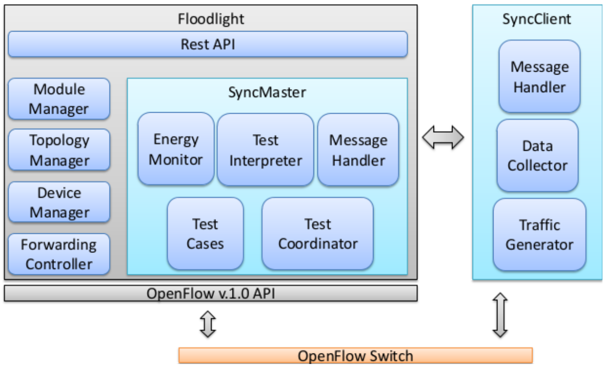


Fig. 3. Modular representation of the system design

average the power consumption.

The DUTs are a hardware and a software switch. The virtual switch is the Open vSwitch version 1.10.2, running on a Linux rack server (Dell PowerEdge R320, Intel Xeon E5-1410, 8GB DDR3 RAM, TOSHIBA MG03ACA2 SSD, I350T4 4x Gbit NIC). To eliminate the influences of the system as far as possible, all services not required by the Open vSwitch were shut down. The physical switch is the OpenFlow enabled NEC PF 5240. It has 48 ports, which can be configured on a line speed of 10, 100 or 1000 Mbps and features a backplane capacity of 176 Gbps or 131 Mpps.

Figure 3 describes the architecture of the power modeling framework, consisting of the *SyncClient* and the *SyncMaster* which is embedded into the Floodlight framework. The *SyncClient* configures the *Traffic Generator* and reads measurements from the *Data Collector*, which are sent to the *SyncMaster* via the *Message Handler*.

More details considering the measurement setup and accuracy of the measurements are detailed in [19] and [20].

IV. Measurements

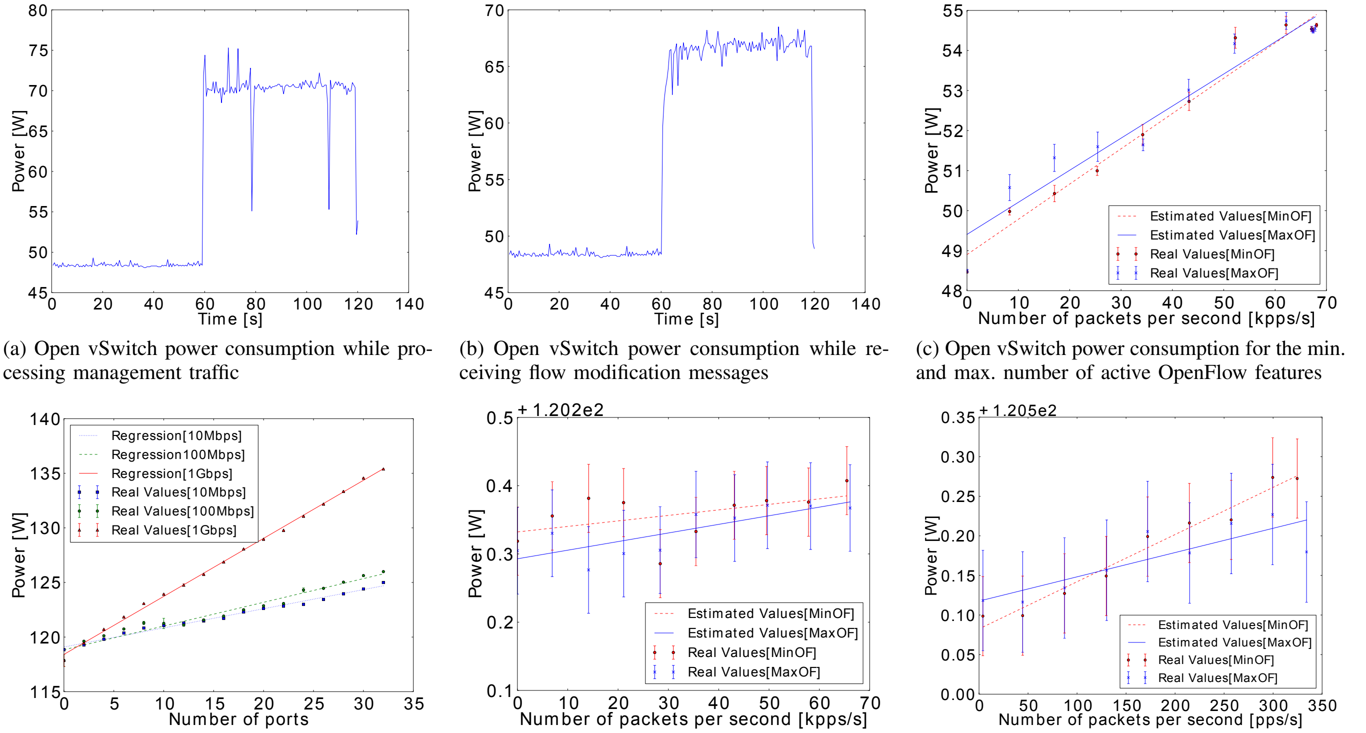
First, the power consumption of the Open vSwitch is evaluated. According to the test dependencies, the dynamic power consumption of the Open vSwitch depends on two types of traffic, namely management traffic, consisting of packet-in and flow modification messages representing the most common and most important control messages, and the OpenFlow traffic. In order to measure the packet-in influence, the traffic generator generates a high traffic throughput (1 Gbps), while the switch has no rules installed at all. As a result, the switch starts requesting rules from the OpenFlow controller. The increase of the power (i.e. the difference) per packet over the idle state yields the packet-in energy in Table I. Similarly, Figure 4b shows a power increase due to flow modification messages. Based on the difference between idle and active power, and the number of messages being sent, the energy needed to install a new OpenFlow rule is calculated.

Similarly, the influence of the OpenFlow controlled traffic on the power consumption is evaluated. In order to inspect if, and how, OpenFlow rules being applied on packets influence the power consumption, the bandwidth of the traffic is successively increased, while having different OpenFlow rules active. Figure 4c shows the correlation between forwarded packets and the power consumption of the Open vSwitch. Besides, it illustrates two different test cases. The first one installs only two OpenFlow rules, matching on the ingress port and applying the forwarding action on packets. This is the minimum set of OpenFlow rules in order to forward traffic. In comparison, also a maximum test case is applied (represented by MaxOF in Figure 4c), installing merely two OpenFlow applying multiple matches and actions representing the maximum possible set of matches and actions. From this, two conclusions can be deduced. First, higher traffic causes an almost linear increase in power. Secondly, their values approach when traffic throughput increases. Thus, it can be concluded that the number of actions and matches within an OpenFlow rule does not significantly influence the power consumption. Performing the same tests with identical packet rate, but smaller packets (i.e. lower bandwidth), leads to a comparable power consumption.

A similar test set is also run on the hardware switch. Still, first influence of the configuration of the available ports is evaluated. Figure 4d shows the relation and fitted approximation ($R^2 = 0.971$) between active port number, configured rate, and power. The respective measurements are given in Table II, showing a 62% energy saving potential when switching 1 Gbps links to 10 Mbps when inactive, or 55% when switching to 100 Mbps.

Next, the performance tests are run on the NEC switch. Initial tests show large deviations in the maximum supported packet rate for different configurations. Hence, the data rates of the test are adjusted to match the limits imposed by the hardware. First the influence of hardware-supported OpenFlow rules is analyzed (cf. Fig. 4e). Secondly, the correlation between software implemented OpenFlow rules is evaluated (cf. Figure 4f). Both figures indicate the small impact of traffic on the power consumption. The increase of power ranges between 100 mW and 200 mW within those test cases and is thereby almost negligible. Furthermore, for both measurement series, there is only a small difference in terms of power consumption between a minimum set of actions and a maximum set within an OpenFlow rule, as indicated by the overlapping confidence interval of the measurements.

The resulting parameters for the different operating states, matches, and actions are summed up in Table I. The main difference between those switches is that the power consumption of the hardware switch is almost independent of the traffic. Although the power consumption of the software switch depends highly on the traffic, there are no significant differences between different combinations of actions and matches. This is confirmed by the Figures 4c, 4e and 4f. Contrary, the power parameters in Table I and the two Figures 4a and 4b clearly show the influence of management packets on the power consumption. A further difference between the software and hardware switch is that the NEC switch consumes higher power per packet if only software supported flows are processed. Software implemented matches and ac-



(a) Open vSwitch power consumption while processing management traffic

(b) Open vSwitch power consumption while receiving flow modification messages

(c) Open vSwitch power consumption for the min. and max. number of active OpenFlow features

(d) NEC power consumption for a varying number of active ports

(e) NEC power consumption while processing hardware accelerated operations

(f) NEC power consumption while processing software supported operations

Fig. 4. Comparison of the power consumption of the NEC PF5240 and the Open vSwitch under different load conditions

tions are detectable in Table I by their power parameters, which are about 1000 times higher than those of hardware supported OpenFlow features. Another interesting observation are the small Open vSwitch energy values (about 0) in Table I, which are caused by the dependencies of test cases (cf. Fig. 2), where the additional match does not induce additional processing overhead.

A more detailed description of the measurements can be found in [19] and [20].

V. Power Consumption Model for OpenFlow Switches

The power models are generated by correlating the power and performance measurements from the previous section. Using the hierarchy of the test cases as defined in Section III, the structure of the power model is defined. Accordingly, the power consumption of the switch P_{switch} is modeled as:

$$P_{\text{switch}} = P_{\text{base}} + P_{\text{config}} + P_{\text{control}} + P_{\text{OF}} \tag{1}$$

It consists of the base power P_{base}, the power of the configuration (i.e. number active ports, configured line speed) P_{config}, the power consumption of the control traffic P_{control}, and the power consumption of the processed OpenFlow traffic P_{OF}.

The configuration power P_{config} depends on the number of active ports and the configured line speed. Hence, the configured power is

$$P_{\text{config}} = \sum_{i}^{N_{\text{activePorts}}} s_i \cdot P_{\text{port}}, \tag{2}$$

where $N_{\text{activePorts}}$ is the number of active ports, s_i is the relative power consumption of the configured speed of the port and P_{port} is the power consumption of the port at full speed. The power of the control traffic

$$P_{\text{control}} = r_{\text{PacketIn}} \cdot E_{\text{PacketIn}} + r_{\text{FlowMod}} \cdot E_{\text{FlowMod}} \tag{3}$$

is defined by the rate of outgoing *PacketIn* messages r_{PacketIn}, the rate of incoming *FlowMod* messages r_{FlowMod}, and their respective energy consumption per packet E_{PacketIn} and E_{FlowMod}. The power consumption of the processed OpenFlow traffic P_{OF} is modeled by

$$P_{\text{OF}} = \sum_{i}^{N_{\text{flows}}} r_{\text{packets}}(i) \left[\sum_{j}^{N_{\text{matches}}} \mu_{\text{match}}(i,j) \cdot e_{\text{match}}(j) \right. \\ \left. + \sum_{k}^{N_{\text{actions}}} \mu_{\text{action}}(i,k) \cdot e_{\text{action}}(k) \right] \tag{4}$$

Here, N_{flows} is the number of active flows with the respective packet rate $r_{\text{packets}}(i)$. For each processed packet, a number of matches and actions are possible. The power consumption of the matches are modeled as the sum over the possible matches j, where active matches are denoted by the binary values $\mu_{\text{match}}(i,j)$, and the energy of a single match is $e_{\text{match}}(j)$. Similarly, the power consumption of the actions is modeled as the sum over the possible actions k, their energy $e_{\text{action}}(k)$ and their use $\mu_{\text{action}}(i,k)$, as binary values.

Considering the highly different power characteristics of the hardware and the software switch, the general model can be

TABLE I
POWER AND ENERGY PARAMETERS OF THE SWITCHES AND THE CORRESPONDING OPENFLOW FEATURES.

	NEC	**OvS**
Base Power [W]		
Base	118.3300	48.7397
Port Configuration Power [W/port]		
Port	0.5295W	n/a
Port Configuration Power Factor		
Line Speed [1Gbit]	1	1
Line Speed [100Mbit]	0.4455	n/a
Line Speed [10Mbit]	0.3761	n/a
Management Energy [μWs/packet]		
Packet-In	711.3028	775.5246
Flow-Modification	20.2512	1,445.1309
Energy for Matching [μWs/packet]		
Ingress Port	1.1013	47.1955
MAC Src	0.4234	51.4126
MAC Dst	0.7286	50.1219
IP Src	0.3982	50.4271
IP Dst	0.4390	48.3222
L4 Port Src	0.6725	0
L4 Port Dst	974.1202	0.5716
Energy for Actions [μWs/packet]		
Output	0.4286	49.1954
MAC Src	0	0
MAC Dst	0.5672	0
IP Src	728.2008	0.5119
IP Dst	866.8180	0.8716
L4 Port Src	977.2340	5.1381
L4 Port Dst	846.8505	0.9332

simplified. Considering the minimal effect of the processed traffic on the NEC switch, a traffic independent power model seems reasonable. For this, the traffic dependent term in Equation 1 is set to 0, from which the simplified equation for the power consumption of the hardware switch follows

$$P_{\text{switch,HW}} = P_{\text{base}} + P_{\text{config}} + P_{\text{control}}. \qquad (5)$$

Similarly, the traffic processed by the Open vSwitch and its effect on the power consumption is evaluated. As already visible in Figure 4c, the difference between the minimum and maximum number of matches and actions applied to a packet is minimal. Hence, a maximum power model is proposed. From the energies for matches and actions applied to each flow (cf. Equation 4), the maximum energy of the active matches and actions is selected and applied to each flow. Hence, the equation of the software switch the influence of the processed traffic is simplified to

$$P_{\text{OF,SW}} = \sum_{i}^{N_{\text{flows}}} r_{\text{packets}}(i) \cdot \Big(\max_j \left[\mu_{\text{match}}(i,j) \cdot e_{\text{match}}(j)\right] + \max_k \left[\mu_{\text{action}}(i,k) \cdot e_{\text{action}}(k)\right] \Big). \qquad (6)$$

VI. EVALUATION OF THE DERIVED POWER MODELS

For the evaluation of the power models, the Floodlight plugin is extended by the *Energy Monitor* component to allow real-time monitoring of the power meter, the power estimates, and the error between both using different models for the DUT. Therefore, the Floodlight plugin connects to the measurement client (cf. Figure 1), while estimating the power consumption of the DUT based on the configuration, processed traffic, and management traffic. Figure VI shows power consumption and the resulting error for increasing traffic on the Open vSwitch, which is visible in the top graph. The second and third graph show the hardware power measurement and the model based power estimation, both closely following the traffic pattern. The last graph shows the error between the measurement and the power estimate. The error is generally lower than 3W, relating to a root mean square error (RMSE) of 5% compared to the maximum power consumption of the switch.

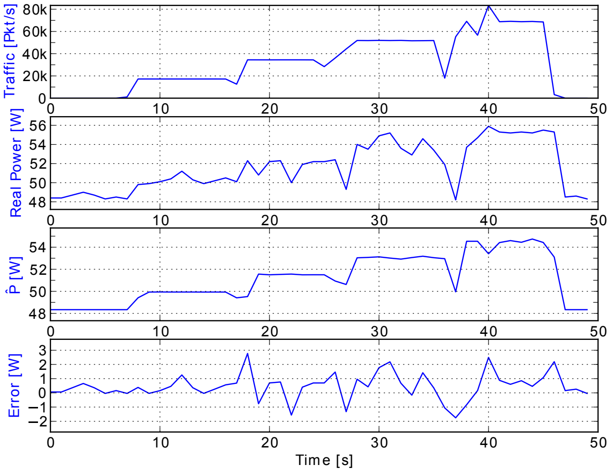


Fig. 5. Assessment of the accuracy of the power measurements and model based estimation of the Open vSwitch for varying load

TABLE II
ERROR OF THE DERIVED POWER MODELS

Switch	**NEC PF 5240**			**Open vSwitch**	
Model	**general**	**maximum**	**static**	**general**	**maximum**
$P_{e,\text{max}}$ [W]	7.56	1	0.72	9	4.30
$P_{e,\text{rel,max}}$	5.6%	0.7%	0.5%	16.1%	7.7%

The accuracy and the resulting error observed during further test of different power models on both devices is given in Table II. These are detailed in [19] and [20]. The table compares the general and simplified power model as described in Section V. The maximum observed error $P_{e,\text{max}} = \max(|P - \hat{P}|)$ gives the upper limit for the errors. The maximum relative error, which is identical to the accuracy of the model, is given by $P_{e,\text{rel,max}} = |P_{e,\text{max}}/P|$. Generally, the power consumption using the simplified models can be approximated with an error lower than 1% for the hardware switch and less than 8% of the software switch.

VII. DISCUSSION AND CONCLUSION

In the beginning, this paper posed three questions. First, the cost of the OpenFlow rule-set on switching and routing decisions was questioned. Section IV shows that for the Open vSwitch additional matches or actions on the same packets don't increase the power consumption. On the hardware switch, the additional power consumption caused by traffic is barely measurable, as long as all operations are hardware supported. Still, the per packet power consumption is 1000-fold while processing not accelerated matches and actions in the CPU of the NEC switch.

The second question targeted the performance differences between hardware and software implementations of the OpenFlow rule-set. The performance of the hardware implemented features of the NEC switch is by a factor of 300.000 higher (130 Mpps instead of 350 pps). Contrary, the measurements of the Open vSwitch were limited by the bandwidth of the (single) network link usable for productive traffic, limiting the packet rate to 70 kpps, which is by a factor of 200 higher than software supported features in the NEC device, but dimensions away from the (theoretical) 130 Mpps of the hardware supported features.

Thirdly, the improvement of the energy-efficiency of the analyzed devices was questioned. Concluding from the observed behavior and the derived power models of the Open vSwitch, aggregation of traffic and aggressive sleeping policies should increase the energy efficiency. Considering the small influence of the processed traffic on the power consumption of the hardware OpenFlow, and the large influence of the number of active interfaces and their configuration on the power consumption, the switch configuration is the most promising area for optimization. Reducing the configured rate on all ports to 10 Mbps allows saving of up 62% of the interface power consumption, relating to an overall power reduction of 6.6% without interrupting connectivity.

Concluding, three distinct contributions were made. First, a power measurement and model generation framework based on Floodlight was introduced. Based on these power models, a Floodlight plugin is described allowing the real-time power estimation of OpenFlow enabled switches connected to the OpenFlow controller, given that a power model for the respective device type was calibrated or is available otherwise. Thirdly, the general behavior of OpenFlow hardware and software switches was evaluated based on the derived power models. From this, the possible improvements concerning the configuration and traffic management were derived. The resulting accuracy of the described power models is 8% for the software and less than 1% for the hardware switch.

Possible future work includes the extension of the framework to support OpenFlow versions above 1.0. Further, more OpenFlow devices need to be measured and the power models published. This allows estimating the power consumption of heterogeneous networks without calibrating power models. For pure software switches, the system load should be monitored, and the system utilization correlated with the processed flows. This way, previously generated (server/PC) power models may be used to estimate the power consumption of the Open vSwitch, extending the usability of the generated models. Using the developed power models, routing and links can be adjusted as suggested in [11], [15], [16], by switching off interfaces or re-routing traffic via the most energy efficient links. Finally, the dynamic energy consumption patterns of candidate devices can also provide a basis for network design and purchasing decisions.

Acknowledgments

This work has been supported in parts by the European Union (FP7/#317846, SmartenIT and FP7/#318398, eCOUSIN) and the DFG as part of the CRC 1053 MAKI. The authors would like to acknowledge valuable comments by their colleagues and project partners.

References

[1] B. Raghavan and J. Ma, "The Energy and Emergy of the Internet," in *ACM HotNets*, 2011.

[2] V. C. Coroama and L. M. Hilty, "Assessing Internet Energy Intensity: A Review of Methods and Results," *Elsevier Environmental Impact Assessment Review*, Vol. 45, pp. 63–68, Feb. 2014.

[3] B. Heller, S. Seetharaman, P. Mahadevan *et al.*, "ElasticTree: Saving Energy in Data Center Networks," in *USENIX Symposium on Networked Systems Design and Implementation*, 2010.

[4] M. Charalambides, D. Tuncer, L. Mamatas, and G. Pavlou, "Energy-Aware Adaptive Network Resource Management," in *IFIP/IEEE IM*, 2013.

[5] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR)," *IEEE Transactions on Computers*, Vol. 57, No. 4, pp. 448–461, 2008.

[6] N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN," *ACM Queue*, Vol. 11, No. 12, pp. 20–40, Dec. 2013.

[7] N. McKeown, T. Anderson, H. Balakrishnan *et al.*, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, Vol. 38, No. 2, pp. 69–74, 2008.

[8] S. Jain, M. Zhu, J. Zolla *et al.*, "B4: Experience with a Globally-Deployed Software Defined WAN," in *ACM SIGCOMM*, Aug. 2013.

[9] H. Hlavacs, G. Da Costa, and J.-M. Pierson, "Energy Consumption of Residential and Professional Switches," in *IEEE CSE*, 2009.

[10] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A Power Benchmarking Framework for Network Devices," in *IFIP Networking*, 2009.

[11] Y. Yang, D. Wang, M. Xu, and Suogang Li, "Hop-by-hop Computing for Green Internet Routing," in *IEEE ICNP*, 2013.

[12] J. Chabarek, J. Sommers, P. Barford *et al.*, "Power Awareness in Network Design and Routing," in *IEEE INFOCOM*, 2008.

[13] P. T. Congdon, P. Mohapatra, M. Farrens, and V. Akella, "Simultaneously Reducing Latency and Power Consumption in OpenFlow Switches," *IEEE/ACM Transactions on Networking*, 2013.

[14] C. Rotsos, N. Sarrar, S. Uhlig *et al.*, "OFLOPS: An Open Framework for OpenFlow Switch Evaluation," in *PAM*, 2012.

[15] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing ISP Network Energy Cost: Formulation and Solutions," *IEEE/ACM Transactions on Networking*, Vol. 20, No. 2, pp. 463–476, 2012.

[16] N. Vasic, D. Novakovic, S. Shekhar *et al.*, "Identifying and Using Energy-Critical Paths," in *ACM CoNEXT*, 2011.

[17] R. Bolla, R. Bruschi, F. Davoli *et al.*, "A Northbound Interface for Power Management in Next Generation Network Devices," *IEEE Communications Magazine*, pp. 149–157, Jan. 2014.

[18] B. Pfaff, "Openflow Switch Specification Version 1.0.0," OpenFlow Switch Consortium, Tech. Rep., 2009.

[19] F. Kaup, S. Melnikowitsch, and D. Hausheer, "A Model-based OpenFlow Power Monitoring Framework based on Real-World Measurements," PS Technical Report No. PS-TR-2014-02, Tech. Rep., 2014. http://www.ps.tu-darmstadt.de/fileadmin/publications/PS-TR-2014-02.pdf

[20] S. Melnikowitsch, "Measuring and Modeling the Power Consumption of Software Defined Networks," Master's Thesis, Technische Universität Darmstadt, 2014.