

We've Got the Power: A Framework for Real-Time Network Power Monitoring

Rahil Gandotra¹, Levi Perigo²

¹Interdisciplinary Telecom Program, University of Colorado Boulder, Boulder, USA

²Department of Computer Science, University of Colorado Boulder, Boulder, USA

Email: rahil.gandotra@colorado.edu, levi.perigo@colorado.edu

How to cite this paper: Gandotra, R. and Perigo, L. (2020) We've Got the Power: A Framework for Real-Time Network Power Monitoring. *Journal of Computer and Communications*, 8, 75-88.

<https://doi.org/10.4236/jcc.2020.85004>

Received: April 16, 2020

Accepted: May 25, 2020

Published: May 28, 2020

Copyright © 2020 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The energy consumption of the information and communication technology sector has become a significant portion of the total global energy consumption, warranting research efforts to attempt to reduce it. The pre-requisite for effectual energy management is the availability of the current power consumption values from network devices. Previous works have attempted to estimate and model the consumption values or have measured it using intrusive approaches such as using an in-line power meter. Recent trends suggest that information models are being increasingly used in all aspects of network management. This paper presents a framework developed for enabling the collection of real-time power consumption information from the next generation of networking hardware non-intrusively by employing information models. The experiment results indicate that it is feasible to gather power consumption data using standardized IETF information models, or non-standard customized information models, or through abstracting and exposing the information in a uniform format when no support for the required information models exists. Functional validation of the proposed framework is performed and the results from this research could be leveraged to make energy-efficient network management decisions.

Keywords

Energy Management, Real-Time Monitoring, Information Models, Python, MIB, YANG, API

1. Introduction

The Information and Communication Technology (ICT) services play an indispensable part in our lives and have transformed numerous activities including economic productivity, health, education, democracy, and sustainable develop-

ment. The ICT infrastructure comprises of devices belonging to the consumer end, the network infrastructure, and inside the data centers. The energy consumption of this infrastructure has been rising steadily from 1815 TWh or 8% of the global energy use in 2012, to 2547 TWh or 10% of the global energy use in 2017 [1]. These energy consumptions translate into large carbon footprints since most of the energy produced globally still comes from burning fossil fuels, which is a carbon-intensive approach to energy production [2]. Researchers have predicted the global energy demand of the ICT sector by 2030 to be 8265 TWh or 21% of the global energy use, with the worst-case prediction being 30,715 TWh or 51% of global energy use [3]. The network infrastructure alone is predicted to consume 2641 TWh by 2030 making it a considerable portion of the total ICT energy consumption. These numbers indicate that efforts to reduce energy consumption in computer networks are necessitated.

The ISO/ITU network management model consists of five areas of concern: fault, configuration, accounting/administration, performance, and security (FCAPS). However, the criticality of energy management warrants its inclusion in the traditional definition of network management. Energy management refers to the activities, methods, procedures and tools that enable the planning, measurement, modeling, control, and optimization of the use of energy in networked systems [4]. The goal of energy management is to reduce the network energy consumption and thereby reduce carbon emissions resulting from the ICT infrastructure [5]. To develop energy-saving management policies, the foremost requirement is that of pertinent operational energy data be available from the network infrastructure. This involves the power/energy consumption information of a device and its components, which allows for understanding the relationship between a device's power consumption and its operating configurations. Therefore, any energy management framework should include the capability to read and collect the current power consumption information from devices to aid in energy management. We use the terms power and energy interchangeably in this paper.

Typically, a device specification data sheet mentions only its nameplate power, *i.e.* the maximum power drawn by the device. Since the actual power consumed by network devices depends on various factors such as device configurations and traffic workload, relying only on the specified nameplate power can grossly overestimate the actual device power consumption values [6] [7]. While many present-day networks are not equipped with internal power instrumentation to provide information on the devices' energy consumption at any given time, we expect the next generation of network equipment to have more support for measuring this information. Similar to the majority of the consumer equipment which has battery meters, the reduced cost of power metering instrumentation would allow network equipment vendors to provide this capability.

The motivation behind this work is that if the actual device power consumption information can be collected and made available in a real-time manner, it would allow for better decision making while designing and testing ener-

gy-aware management policies [8]. The results from this research could be utilized to understand devices' power consumption, efficiency, and proportionality behaviors. The rest of the paper is organized as follows: the primary research question and the associated sub-problems are defined in Section II, Section III reviews existing work and briefs the novelty of this research. Section IV describes the research methodology including the environment and technologies incorporated. Section V provides the research results and analysis, and Section VI concludes this research and discusses future work.

2. Research Objectives

The primary research question answered in this paper is **can real-time power data be gathered from network devices non-intrusively**. The following sub-problems were developed to effectively answer the primary research question.

- 1) Can a standardized IETF information model be employed to retrieve power information from network devices?
- 2) Can a customized information model be employed to retrieve power information from network devices?
- 3) Can the power information be collected by management systems which do not support information models?

3. Literature Review

This section describes the prior work performed on modeling and measuring the power consumption of network devices and their components, and sets our approach in context with the related work. Much of the previous work in the area of power consumption of network devices has been focused on estimating it using different methods such as developing statistical traffic models [9] or various analytical models [10] [11] [12] [13] [14]. In [15], Orgerie *et al.* proposed an end-to-end cost model and a simulator for evaluating power consumption in large-scale networks. The developed model computed the energy consumption per equipment while assuming that the power characteristics of the devices are known. The simulator module is integrated with NS2, and with NS3 in [16], and takes user inputs such as the network topology, the network traffic, and the energy consumption values of the network equipment used.

Chabarek *et al.* measured the power demand of two widely used routers over a range of configurations [17]. A digital multimeter equipped with an AC current clamp was used to measure power, and the authors used the combined set of measurements to formulate a general model for router power consumption. While this approach allowed for measuring the real-time consumption values of network devices, measuring in a production network with thousands of devices would be infeasible. The authors extended their prior work in [18] to develop a network-wide power consumption framework. The work involved employing an application programming interface (API) to interface with existing network

management tools, comparing the device configurations with the community database benchmark measurements, and auditing to infer any missing data to enable power consumption estimations. The current running snapshot of a device is associated with power consumption measurements made in a controlled benchmark environment, which makes incorporating the numerous configuration and operational variables that affect the power consumption of devices into the community database non-trivial.

Mahadevan *et al.* presented a power measurement study of a variety of network devices such as hubs, edge switches, core switches, routers, and wireless access points [6]. The benchmarking process involved unplugging the network device from its power outlet and plugging it into the receptacle of a digital power meter. The observed measurements are then used to develop a linear model to capture the total power consumption of switches/routers. In [19], the authors measured the impact of network load, QoS management, and BGP updates on the power consumption of a router by using a wattmeter. Kaup *et al.* studied the cost of the OpenFlow rule set on switching and routing functions, and the difference in performance of the hardware and software implementations of OpenFlow [20]. The measurement set up involved plugging the device under test to a power meter to evaluate the effect of configuration, management and the managed traffic on the resulting power consumption. While it is possible to unplug network devices from their power outlets in a research or academic setting, this disruptive approach would be impractical in an operational network.

The European Telecommunications Standards Institute (ETSI) published a standard, ES-203 237, the Green Abstraction Layer, that specified the Green Standard Interface (GSI) for a uniform way for interactions between the energy-aware hardware and the control framework [21]. The proposed GSI intended to provide the functionalities of discovery, control plane retrieving information about the different energy states supported by the data plane, and monitoring, exchanging relevant device parameters. The goal of this standard was to represent an abstraction of the energy-aware capabilities of networking devices to higher-layer protocols. The Internet Engineering Task Force (IETF) published the Energy Management Framework (eMAN) standard [4], which presented a physical reference model and an information model for devices and device components within, or connected to, networks. The framework modeled relationships and capabilities between energy objects such as power, power state, energy, demand, power attributes and battery. The Society of Cable Telecommunications Engineers' (SCTE) Energy Management Subcommittee (EMS) published a standard ANSI/SCTE 216 2015-Adaptive Power System Interface Specification (APSIS) [22]. They proposed interfacing APSIS applications with the network elements over standard management protocols such as SNMP, NETCONF, IPDR, and HTTP. While these standards described the high-level possible approaches which could be undertaken to measure power in real time, we find that the primary research question defined in this paper still remains unanswered.

Based on the review of the existing body of knowledge in this field, there has been limited work performed on the design and development of a real-time power monitoring framework for the network hardware of the future. The novelty of this research is that we attempt to utilize existing technologies in order to enable the measurement of power consumption values of network devices which have internal power metering capabilities. The proposed framework aims to collect this information non-intrusively—without having to unplug the network device and thereby causing a disruption—from actual production networks. To ensure this work is holistic, the proposed framework considers the power information available through different datastores on the device, transported over the currently used management protocols, encoded in various schemes.

4. Research Methodology

This section provides details on the proposed power monitoring framework, the custom applications developed for the experiment, and the technologies used in this research. **Figure 1** depicts the high-level functional diagram of the real-time network power monitoring framework.

The fundamental assumption behind this work is that the physical network device has the capability to meter and expose its power consumption data through an information data store. As energy management has become critical for the ICT sector, we expect an increasing number of new networking hardware developed to include this capability. The work in this paper is a novel approach and validation of a distributed systems problem of transferring specific data between different entities. The proposed framework aims to gather the power information exposed by the network devices in different formats, using a variety of management protocols and encoding schemes. In order to make this data available for use by other management applications, the proposed framework also includes a northbound API. Additionally, to enable better decision-making, the proposed framework could also interface with an existing network management system (NMS) in order to correlate data between different variables and allow for holistic network management. Next, we describe how the network power monitoring application was designed.

To answer the first sub-problem (see **Figure 2**), we employed IETF standard energy management information bases (EMAN MIBs) defined using the mechanisms specified in the Structure of Management Information (SMIv2), communicated using the SNMP protocol [23]. The conceptual model describes three classes of energy objects—device, component, and power interface. The EMAN MIB contains the eoPowerTable that provides the capability to report power consumption for each energy object. Devices and their components are modeled and identified using the containment tree described in the Entity-MIB [24]. The goal of this sub-problem was to validate if the collection of power information exposed using standardized models is achievable.

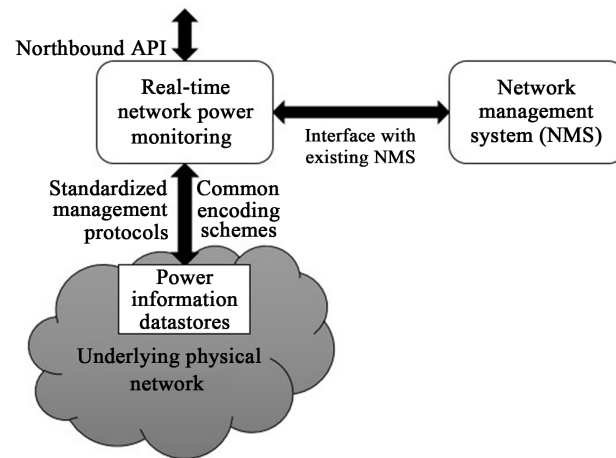


Figure 1. High-level real-time network power monitoring framework.

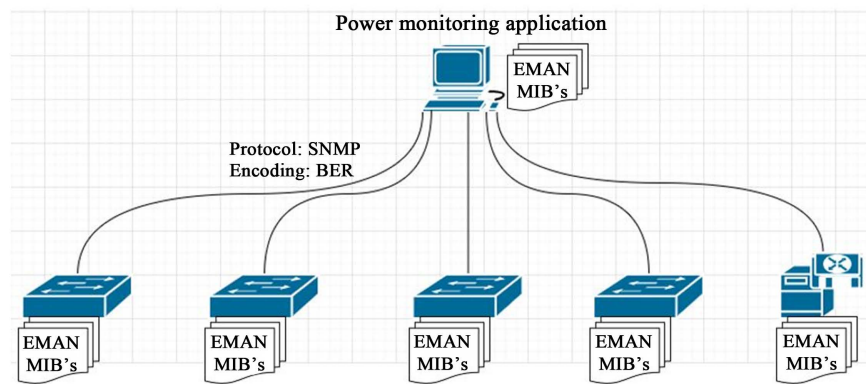


Figure 2. Using IETF standard models.

To answer the second sub-problem (see **Figure 3**), we used custom YANG modules which define the power and energy information of network devices using the YANG data modeling language [25]. OpenConfig is an informal working group of operators that develops vendor-neutral YANG modules aimed to be independent of the data encoding, transport, or RPC protocols [26]. The supported transport protocols include NETCONF, RESTCONF, GRPC and GNMI, and the possible encoding schemes include XML or JSON. The objective of this sub-problem was to verify if the power information exposed using a customized information model, encoded in different schemes, and transported over different protocols can be collected.

To answer the third sub-problem, the power monitoring application developed (see **Figure 4**) saves the collected information in a database. A simple database schema is used to store the information gathered in different formats from different sources in a common structured manner. To make this stored information available to other management systems running in the network, REST endpoints are initialized in order to enable application-to-application communication. The endpoints consist of a common HTTP endpoint to fetch power information of all the devices, and device-specific HTTP endpoints identified by

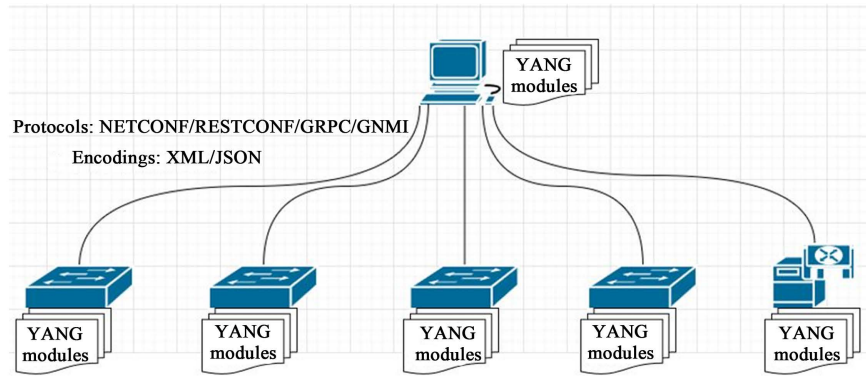


Figure 3. Using custom information models.

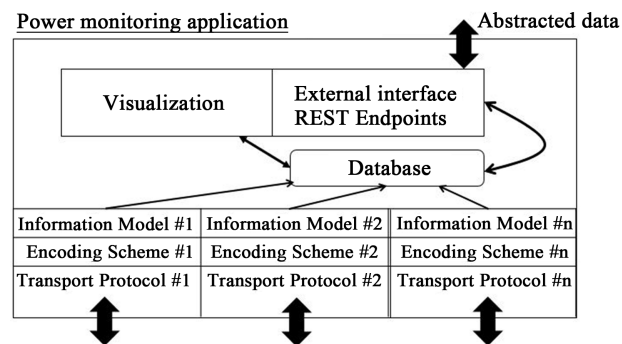


Figure 4. Power monitoring application.

the device's remote management endpoint (IP:port). The aim of this sub-problem was to test the collection of power data by another management system or a user having no support of the information models and transport protocols needed to communicate with the network devices.

Since none of the devices present in the network engineering research laboratory at the University of Colorado Boulder had internal power metering instrumentation, we utilized the power consumption values of five different network devices from our previous research [27] while simulating network agents in this work. The simulation setup is shown in **Figure 5**.

The network agents were simulated on a x86_64-bit server with an Intel Xeon E3 Core i5 processor having 8 CPU cores, Gigabit interfaces, and 128 GB of RAM running the Ubuntu 16.04.5 server operating system. For the first sub-problem, the network agents were simulated using a Python-based, open-source implementation of SNMP agents [28]. For the experiments, five SNMPv3 agents were simulated listening on UDP port numbers 1024-1028. The simulated agents responded to SNMP requests based on the associated simulation data files. The simulation data files consisted of pipe-separated triplets of OID, type, and value. For this work, the OIDs identified belonged to the Entity and EMAN MIBs. For the second sub-problem, YANG agents were simulated from a customized energy management YANG module based on the EMAN MIB [29]. Five agents were simulated on the server listening on TCP port

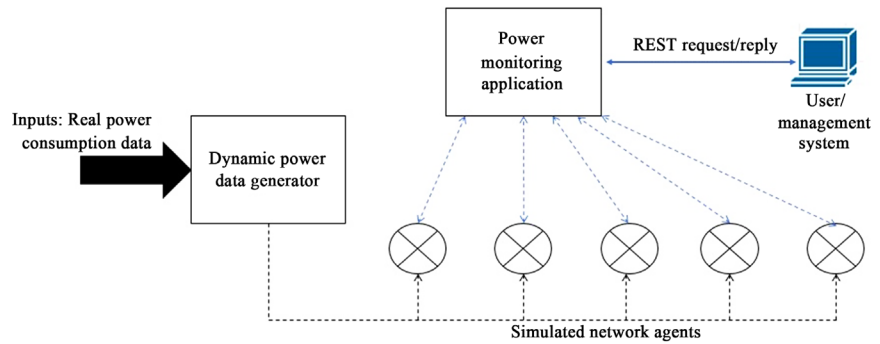


Figure 5. Simulation setup.

numbers 8830-8834. The simulation data files consisted of comma-separated triplets of leaf nodes, type, and value.

The dynamic power data generator application was developed in Python and was responsible for generating randomized time-series power consumption data (in Watts) from the real measured data from previous work. Upon generating the randomized power data for the five agents, the application edited the simulation data files for each agent to modify the value of the specific OID/leaf node that represented the current power consumption of the network device. This enabled the simulation of live network devices with fluctuating power consumption value for the experiments, and to check the real-time collection functionality of the framework.

The power monitoring application was written in Python and consisted of four modules—discovery, retrieval, visualization, and external interface. The discovery module was responsible for discovering the network endpoints to be monitored through a Network Source of Truth (NSoT) file or through manual inputs by a user. The information needed to monitor included the device’s management IP, the supported information model, the supported transport protocols, the supported encoding schemes, the layer-4 port number the device is listening on, and the credentials required to establish the transport session. This included the community string, username, and authentication and privacy protocols’ passphrases for SNMP, and username and password for NETCONF/RESTCONF. Next, the discovery module attempts to establish management sessions with the agents and determines whether they have the capability to monitor and expose their power consumption information. For the agents that do, the discovery module then ascertains the index number that identifies the entire device chassis, since we are interested in monitoring the power consumption of the whole device and not its components. However, the application has the capability to discover and monitor components-level power consumption information.

The retrieval module was responsible for fetching data on the measured power and the direction of power flow. Net-SNMP was employed to fetch data from SNMP agents [30], ncclient was employed for NETCONF agents [31], and Py-RESTCONF for RESTCONF agents [32]. The module was developed to be mul-

ti-threaded in order to enable the simultaneous collection of data from multiple agents. Next, the retrieval module collected the values fetched, connected to a local database, and updated the specific device rows with the current power information. The visualization module was responsible for creating a dynamic image to communicate the real-time power information. NetworkX was used to visualize the current network topology with the node attributes being the real-time power consumption of devices fetched from the database [33]. The external interface module was responsible for making the stored information available to other management systems or users by creating HTTP endpoints. Flask-RESTful was employed to initialize the REST API on TCP port number 5002, and different class objects were declared to identify the entire information in the database, as well as specific device information [34].

5. Results and Analysis

In this section, the experiment results for each of the subproblems defined in Section 2 are presented and analyzed to effectively answer the primary research question also defined in Section 2.

5.1. Can a Standardized IETF Information Model Be Employed to Retrieve Power Information from Network Devices?

To test this sub-problem, the power monitoring application developed first discovered the five simulated network agents from the NSoT file which contained the information required to connect to the agents using SNMP—agent IP and port, SNMP version, security level, authentication protocol, authentication key, security name, encryption protocol, and encryption key. The application then discovered the value of `entPhysicalIndex` which identifies the device chassis index in order to collect power data of the complete device and not of any specific component. For this, a `snmpwalk` was performed on the `entPhysicalDescr` OID from the ENTITY MIB to find the chassis index value. Next, a SNMP Get operation was performed on the `eoMeterCapability` OID from the EMAN MIB to check if the device supports power monitoring. In this work, the network agents are simulated to have this capability and therefore return True values. Thereafter, the application fetched values from the `eoPower` and `eoPowerUnitMultiplier` OIDs that indicate the power measurement and the magnitude of watts for the usage value. The sign on the `eoPower` value helped determine if the agent was a consumer of power or a producer of power.

The application then updated the database with the current power consumption values and updated the visualized image to reflect the real-time power measurements (see Figure 6). Additionally, the application also supports the collection of device energy consumption data based on specified time intervals. The experiment results helped verifying that if the underlying network hardware has the capability to measure its power consumption, and that information is exposed using a standardized information model such as a SNMP MIB, collection of this information in a real-time manner without causing any network

Key	Value	Key	Value	Key	Value
1 Device Name	Device A	1 Device Name	Device B	1 Device Name	Device C
2 Device Endpoint	127.0.0.1:1024	2 Device Endpoint	127.0.0.1:1025	2 Device Endpoint	127.0.0.1:1026
3 Power monitoring capability	True	3 Power monitoring capability	True	3 Power monitoring capability	True
4 Power in Watts	102.90	4 Power in Watts	40.69	4 Power in Watts	93.28
5 Power Unit Multiplier	0	5 Power Unit Multiplier	0	5 Power Unit Multiplier	0
6 Direction of power	Consumer of power	6 Direction of power	Consumer of power	6 Direction of power	Consumer of power

Key	Value	Key	Value
1 Device Name	Device D	1 Device Name	Device E
2 Device Endpoint	127.0.0.1:1027	2 Device Endpoint	127.0.0.1:1028
3 Power monitoring capability	True	3 Power monitoring capability	True
4 Power in Watts	56.12	4 Power in Watts	39.38
5 Power Unit Multiplier	0	5 Power Unit Multiplier	0
6 Direction of power	Consumer of power	6 Direction of power	Consumer of power

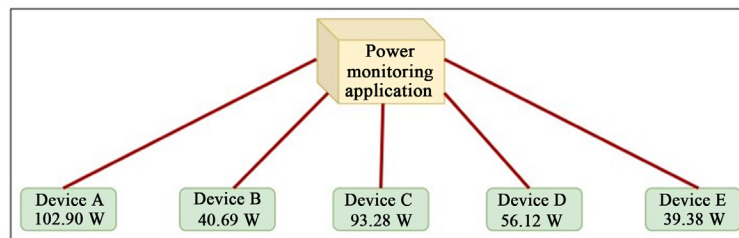


Figure 6. Power data obtained from IETF standardized model.

disruption is achievable. Security is considered by using the user-based security model enabled by SNMPv3.

5.2. Can a Customized Information Model Be Employed to Retrieve Power Information from Network Devices?

Since an increasing number of network operators are moving away from MIB-based modeling to newer network modeling paradigms, the proposed framework must validate its operation when using customized information models. To verify that the proposed framework works independently of different transport protocols and encoding schemes, two experiments were performed. First, the NETCONF protocol with YANG data encoded in XML format was tested, and second, the RESTCONF protocol with YANG data encoded in JSON format was tested.

The power monitoring application first discovered the YANG agents from the NSoT file consisting of the endpoints and credentials information. Since the customized YANG model employed was derived from the EMAN MIB, the procedure to identify the device chassis remained the same, wherein RPC requests (in case of NETCONF) and HTTP requests (in case of RESTCONF) were sent to the entPhysicalDescr leaf list from the ENTITY YANG model to find the chassis ID. Next, the application fetched values from the eoPower and eoPowerUnit-Multiplier leaf nodes by employing NETCONF <get> and RESTCONF GET operations. The application then updated the database with the current power consumption values and updated the visualized image to reflect the real-time power measurements.

Although network modeling is intended to be generic across the Internet eco-

system, specific requirements lead operators to develop customized information models for their infrastructure. The experiment results helped validate that if the power consumption information is exposed using non-standard custom models, encoded in different schemes, and transported over different protocols, the proposed framework exhibits the capability to collect it. Security is enabled by relying on the underlying secure transport protocols—SSH and TLS.

5.3. Can the Power Information Be Collected by Management Systems Which Do Not Support Information Models?

Since different network equipment manufacturers could meter and expose the power data from the devices in several different ways, the collection of this data by a management system not compatible with the devices' management capabilities needed to be tested. If a management system does not support any of the information models, the transport protocols, or the encoding schemes that the devices employ, the collection of power data would be unsuccessful. The developed power monitoring application exposed the power data collected in the database through REST endpoints. An HTTP request for the `/devices` URL to the IP address where the application is hosted results in a response containing the current power information on all devices, while an HTTP request for the `/devices/<device management IP:port>` results in the current power information of the specific device being returned (see [Table 1](#)).

The proposed framework enables abstraction by storing and exposing the data obtained from different sources and in different formats in a homogeneous format. This allows management systems or users to collect power data without having any knowledge of the underlying capabilities required to establish management sessions. Security is provided through restricting REST API access to authorized applications only using HTTP basic access authentication (username/password).

6. Conclusions

Effective network energy management requires real-time power consumption data from the devices in order to incorporate energy awareness into network operations. Owing to reduced costs of power metering and increased criticality

Table 1. Power monitoring application—REST API.

REST URL	Response
<code>/</code>	"REST endpoints present at: EMS-IP:5002/devices, and EMS-IP:5002/devices/<endpoint>"
<code>/devices</code>	<pre>[[["127.0.0.1:1024", "DeviceA", "102.90", 0.0, "consumer"], ["127.0.0.1:1025", "DeviceB", "40.69", 0.0, "consumer"], ["127.0.0.1:1026", "DeviceC", "93.28", 0.0, "consumer"], ["127.0.0.1:1027", "DeviceD", "56.12", 0.0, "consumer"], ["127.0.0.1:1028", "DeviceE", "39.38", 0.0, "consumer"]]]</pre>
<code>/devices/127.0.0.1:1024</code>	<pre>[[["127.0.0.1:1024", "DeviceA", "102.90", 0.0, "consumer"]]]</pre>

of network energy management, it is expected that an increasing number of the next generation of networking hardware will include the capability to measure their power consumption. This research presents a framework for real-time network power monitoring through a non-intrusive approach. Network agents with power measuring capabilities are simulated and an application is developed to test the collection of power data exposed from the devices in different contexts. The first sub-problem analyzed the possibility of collecting power information using IETF standardized information models. The experiment results indicate that an SNMP MIB can be used to gather that information in real time. The second sub-problem tested the feasibility of collecting power information using customized information models, and the testing results confirmed that power data modeled in YANG modules can be obtained using NETCONF or RESTCONF encoded in XML or JSON formats. The final sub-problem aimed to check the collection of data by management systems which do not support information models, and the proposed framework exhibited the capability to abstract and provide the information collected from various sources in a uniform format. This work aims to further the research efforts into ultimately including energy monitoring as a critical part of network management.

The future work would involve analyzing and utilizing the data which made available by the framework proposed in this research to make energy-efficient decisions. A forwarding scheme would be developed that interacts with the power monitoring application to incorporate energy efficiency of individual network hardware into its forwarding decisions.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Corcoran, P. and Andrae, A. (2013) Emerging Trends in Electricity Consumption for Consumer ICT. Proceedings of Tech. Rep., National University of Ireland, Galway.
- [2] Dudley, B. (2019) BP Statistical Review of World Energy. <https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2019-full-report.pdf>
- [3] Andrae, A.S. and Edler, T. (2015) On Global Electricity Usage of Communication Technology: Trends to 2030. *Challenges*, **6**, 117-157. <https://doi.org/10.3390/challe6010117>
- [4] Parello, J., Claise, B., Schoening, B. and Quittek, J. (2014) RFC 7326: Energy Management Framework. Internet Engineering Task Force, Fremont. <https://doi.org/10.17487/rfc7326>
- [5] Gandotra, R. and Perigo, L. (2020) A Comprehensive Survey of Energy-Efficiency Approaches in Wired Networks. University of Colorado, Boulder.
- [6] Mahadevan, P., Sharma, P., Banerjee, S. and Ranganathan, P. (2009) A Power Benchmarking Framework for Network Devices. *Proceedings of the International Con-*

- ference on Research in Networking*, Berlin, 11 May 2009, 795-808.
https://doi.org/10.1007/978-3-642-01399-7_62
- [7] Bianzino, A.P., Raju, A.K. and Rossi, D. (2010) Apple-to-Apple: A Framework Analysis for Energy-Efficiency in Networks. *ACM SIGMETRICS Performance Evaluation Review*, **38**, 81-85. <https://doi.org/10.1145/1925019.1925036>
 - [8] Gandotra, R. and Perigo, L. (2018) SDNMA: A Software-Defined, Dynamic Network Manipulation Application to Enhance BGP Functionality. *Proceedings of the IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems*, Exeter, 28 June 2018, 1007-1014.
<https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00167>
 - [9] Wassal, A.G. and Hasan, M.A. (2001) Low-Power System-Level Design of VLSI Packet Switching Fabrics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **20**, 723-738. <https://doi.org/10.1109/43.924826>
 - [10] Patel, C.S., Chai, S.M., Yalamanchili, S. and Schimmel, D.E. (1997) Power Constrained Design of Multiprocessor Interconnection Networks. *Proceedings of the International Conference on Computer Design VLSI in Computers and Processors*, 12 October 1997, 408-416. <https://doi.org/10.1109/ICCD.1997.628902>
 - [11] Langen, D., Brinkmann, A. and Ruckert, U. (2000) High Level Estimation of the Area and Power Consumption of On-Chip Interconnects. *Proceedings of the 13th Annual IEEE International ASIC/SOC Conference*, 13-16 September 2000, 297-301.
<https://doi.org/10.1109/ASIC.2000.880753>
 - [12] Geethanjali, E., Narayanan, V. and Irwin, M.J. (2002) An Analytical Power Estimation Model for Crossbar Interconnects. *Proceedings of the 15th Annual IEEE International ASIC/SOC Conference*, Rochester, 25-28 September 2002, 119-123.
<https://doi.org/10.1109/ASIC.2002.1158042>
 - [13] Wang, H.S., Peh, L.S. and Malik, S. (2003) A Power Model for Routers: Modeling Alpha 21364 and Infiniband Routers. *IEEE Micro*, **23**, 26-35.
<https://doi.org/10.1109/MM.2003.1179895>
 - [14] Ye, T.T., Micheli, G.D. and Benini, L. (2002) Analysis of Power Consumption on Switch Fabrics in Network Routers. *Proceedings of the 39th Annual Design Automation Conference*, New Orleans, 10-14 June 2002, 524-529.
<https://doi.org/10.1145/513918.514051>
 - [15] Orgerie, A.C., Lefevre, L., Guerin-Lassous, I. and Pacheco, D.M.L. (2011) ECOFEN: An End-to-End Energy Cost Model and Simulator for Evaluating Power Consumption in Large-Scale Networks. *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Lucca, 20-24 June 2011, 1-6. <https://doi.org/10.1109/WoWMoM.2011.5986203>
 - [16] Orgerie, A.C., *et al.* (2017) Simulation Toolbox for Studying Energy Consumption in Wired Networks. *Proceedings of the 13th International Conference on Network and Service Management*, Tokyo, 26-30 November 2017, 1-5.
<https://doi.org/10.23919/CNSM.2017.8256037>
 - [17] Chabarek, J., *et al.* (2008) Power Awareness in Network Design and Routing. *Proceedings of the IEEE INFOCOM 2008 the 27th Conference on Computer Communications*, Phoenix, 13-18 April 2008, 457-465.
<https://doi.org/10.1109/INFOCOM.2008.93>
 - [18] Chabarek, J. and Barford, P. (2013) Energy Audit: Monitoring Power Consumption in Diverse Network Environments. *Proceedings of the International Green Computing Conference Proceedings*, Arlington, 27-29 June 2013, 1-10.

- <https://doi.org/10.1109/IGCC.2013.6604505>
- [19] Adelin, A., Owezarski, P. and Gayraud, T. (2010) On the Impact of Monitoring Router Energy Consumption for Greening the Internet. *Proceedings of the 11th IEEE/ACM International Conference on Grid Computing*, Brussels, 25-28 October 2010, 298-304. <https://doi.org/10.1109/GRID.2010.5697988>
 - [20] Kaup, F., Melnikowitsch, S. and Hausheer, D. (2014) Measuring and Modeling the Power Consumption of Open Flow Switches. *Proceedings of the 10th International Conference on Network and Service Management and Workshop*, Rio de Janeiro, 17-21 November 2014, 181-186. <https://doi.org/10.1109/CNSM.2014.7014156>
 - [21] European Telecommunications Standards Institute (2014) ES 203 237 Green Abstraction Layer (GAL). https://www.etsi.org/deliver/etsi_es/203200_203299/203237/01.01.01_60/es_203237v010101p.pdf
 - [22] Society of Cable Telecommunications Engineers (2015) Adaptive Power System Interface Specification (APSYS). <https://www2.scte.org/SCTEDocs/Standards/SCTE%20245%202018.pdf>
 - [23] Chandramouli, M., Claise, B., Schoening, B., Quittek, J. and Dietz, T. (2015) RFC 7460: Monitoring and Control MIB for Power and Energy. Internet Engineering Task Force, Fremont. <https://doi.org/10.17487/rfc7460>
 - [24] Bierman, A., Romascanu, D., Quittek, J. and Chandramouli, M. (2013) RFC 6933: Entity MIB. Internet Engineering Task Force, Fremont.
 - [25] Bjorklund, M. (2010) RFC 6020: YANG—A Data Modeling Language for the Network Configuration Protocol (NETCONF). Internet Engineering Task Force, Fremont. <https://doi.org/10.17487/rfc6020>
 - [26] OpenConfig (2020) Vendor-Neutral, Model-Driven Network Management Designed by Users. <http://www.openconfig.net>
 - [27] Gandotra, R. and Perigo, L. (2020) Comparing Energy Efficiencies of SDN Hardware Based on Forwarding Configurations. University of Colorado, Boulder.
 - [28] SNMP Labs (2020) SNMP Agent Simulator. <http://snmplabs.com/snmpsim>
 - [29] MG-SOFT Corporation (2020) NETCONF Simulator. <https://www.mg-soft.com/mgNetConfSimulator.html>
 - [30] Net-SNMP (2020) SNMP Implementation. <http://www.net-snmp.org>
 - [31] Ncclient (2020) Python Library for NETCONF Clients. <https://github.com/ncclient/ncclient>
 - [32] PyRESTCONF (2020) A Python RESTCONF API. <https://github.com/lisroach/PyRESTCONF>
 - [33] NetworkX (2020) Software for Complex Networks. <https://networkx.github.io>
 - [34] Flask-RESTful (2020) Simple Framework for Creating REST APIs. <https://github.com/flask-restful/flask-restful>