Energy Audit: Monitoring Power Consumption in Diverse Network Environments

Joseph Chabarek
University of Wisconsin Madison
jpchaba@cs.wisc.edu

Paul Barford
University of Wisconsin Madison
pb@cs.wisc.edu

***Abstract*—Understanding the details of power consumption in networks is a precursor to being able to make effective power management decisions. However, measuring the power consumption is often fraught with difficulty due in part to large and dispersed deployment of heterogeneous equipment. In this paper we present a tool suite called EnergyAudit that is designed for network operators to audit the power consumption in their infrastructure. The tool suite has three components, *(i)* an API that links to existing management infrastructures and enables devices to be queried, *(ii)* a database of benchmark measurements of networking equipment that maps device configurations and operating status to power consumption, *(iii)* an auditing tool that queries the community database. The API simplifies deployment of EnergyAudit. The database enables community and manufacturer contribution of power consumption estimates. The auditing tool can infer missing data and reports both the estimated power consumption for devices in the network and the fidelity of the measurements from which the device-to-power mapping was reported. We describe the details of the tool suite and demonstrate its use in three different network environments with diverse systems.**

I. Introduction

Monitoring the power consumption of systems in medium-scale to large-scale network infrastructures is a challenging problem. The systems in the infrastructure may be geographically distributed across a campus, metro area or around the globe. The systems themselves are often diverse (*e.g.,* routers, switches, and other network appliances), and are typically from a diverse set of manufacturers that use different methods for reporting power levels. However, understanding the energy utilization of deployed systems is vital for managing power costs, designing power efficient network topologies, and strategically deploying new hardware.

Traditionally, switches and routers have been largely traffic agnostic with regards to power consumption [14]. Determining power consumption, even on these relatively static legacy devices is complicated by the fact that most devices do not support accurate power auditing. Worst-case energy usage is often provided for provisioning through the device datasheet or occasionally on the command line, but these estimates can vary drastically from actual consumption. Increasingly, network systems are incorporating energy-aware features such as Energy Efficient Ethernet [7], dynamic link voltage in response to varying cable lengths, and port power-down capability. While these capabilities bode well for progress toward energy proportionality, they call for finer-grained power monitoring and raise new challenges for making effective power management decisions.

978-1-4799-0623-9/13/$31.00 © 2013 IEEE

In this paper we address the problem of auditing the power consumption of systems in networks[1]. The lack of solutions in this space means that network operators must resort to power consumption monitoring at the facility-level or the PoP-level, which precludes device or component-level evaluation. Our perspective is that of an operator that has administrative privileges in their infrastructure. Our objective is to develop the capability to monitor power consumption in widely distributed systems from diverse manufacturers at the device level in a query-response fashion, similar to what is typically done via SNMP.

To address this objective, we present *EnergyAudit*, a tool for measuring and inferring the power consumption of networks and individual networking devices. EnergyAudit has three main components (i) an API for interfacing with previously deployed infrastructure management tools and thereby simplifying adoption, (ii) a community database of benchmark measurements that map device configurations to power consumption, thereby enabling accurate power consumption estimates from device queries and (iii) an auditing component that synthesizes measurements and enables network-wide energy consumption evaluation.

EnergyAudit is designed to flexibly combine novel power auditing capabilities with existing infrastructure monitoring tools currently used by network operators. Examples of widely used tools include Rancid [5], Net-SNMP [9], or custom management scripts, which monitor the uptime or performance of an infrastructure or provide details on configurations and status of devices. By integrating with these tools, we seek to gain access to device details that are sufficient to conduct a power audit without additional monitoring equipment.

EnergyAudit takes device details from the existing infrastructure management software as input and transforms the various formats into a common device description or "snapshot." It then creates an abstraction of the network

[1]We do not consider end hosts in this study, although as will be discussed later, our framework can easily be expanded to accommodate them.

device and attempts to match this against a database of configurations with benchmarked power consumption values. The database currently contains 25 device types and over 75 configurations. Over time, additional configurations can be added by researchers, network operators, or even device manufacturers and thereby enhance the accuracy of power consumption estimates.

Given the large variety of network devices and operating conditions, an audit of deployed systems will likely encounter a device snapshot that does not have a direct match in the benchmark database. To address this, EnergyAudit includes the option of generating an inferred power consumption characteristic for a target device. Our approach is based on either providing a lower bound of power consumption with a similar device present in the database or an interpolation between the two closest device snapshots with respect to function and usage.

We evaluate and demonstrate EnergyAudit by deploying it in three different network infrastructures. The environments include *(i)* a large, state-of-the-art research facility on our campus with diverse networking equipment and advanced monitoring capability, *(ii)* a departmental network that includes a mix of 103 devices, and *(iii)* a large university campus with roughly 6000 devices. We describe the details of how EnergyAudit is used to measure the power consumption in each of these environments, and demonstrate the current coverage of the community database.

The primary contributions of this work are *(i)* a flexible community database for storing and mapping power consumption measurements to devices *(ii)* a method that matches complex network configurations to the benchmark power consumption values in the presence of incomplete data *(iii)* an implementation of an auditing tool that reports network-wide power consumption. EnergyAudit is openly available to the community and currently in use on our campus.

The remainder of this paper is organized as follows. In Section II we provide in-depth detail on the challenges of measuring network power consumption and how they impact our measurement approach. Section III provides a description of EnergyAudit and it's implementation detail. In Section IV we provide details the deployments of EnergyAudit on our campus. We describe related work in Section V. We summarize our study and discuss future work in Section VI.

II. Measuring Power Consumption in Networks

In this section we describe the basic framework for our approach to auditing power in a network, the challenges that must be overcome, and we place our tool in the large community of network management utilities. We describe the general measurement framework that we have developed to address the unique challenges of monitoring power consumption in a network composed of diverse systems.

A. Framework Overview

The starting point for our work is the fact that there are no standard methods or tools for measuring power consumption in diverse operational networks in a fine-grained fashion. We seek to develop a methodology and tools that address this problem that can be easily accommodated into standard network management infrastructures. To that end, we focus on developing an approach that is based on gathering information from devices that may be broadly deployed using standard query-response protocols such as the Simple Network Management Protocol (SNMP) or infrastructure monitoring tools that periodically collect device configuration information.

The fundamental tasks in our approach include gathering a representative network device snapshot, which will include configuration information and possibly dynamic state. We then associate this snapshot with a power consumption measurement that has been made in a controlled benchmark environment. Finally, we present this data pair and/or aggregates of pairs in a way that is meaningful for specific network sites and administrators.

These tasks can be, at times, surprisingly challenging. For example, consider something as simple as an interface name on a device, which is essential in our process of matching a device snapshot to entries in our benchmark database. Within a single management protocol (SNMP) there are potentially a number of names for the field that contains the common interface name *ifname*, *ifalias*, or *ifdescription*. Similarly, the name described on a device command line interface can be different than the name reported by SNMP. There can also be significant variation in conventions between vendors and between different measurement protocols.

Fortunately, many of these challenges have already been addressed by network operators by either adapting standard tools or developing customized scripts. We seek to leverage these existing capabilities and increase the utility of our tools, by building an interface that enables straightforward integration.

We deal with the remaining elements of device name complexity by requiring administrators to import specific device attributes into a naming schema we developed for EnergyAudit. This schema is the basis for storing power consumption measurements in the community database. We have already implemented import libraries for the widely used Rancid [5] management tool, and have developed a partial implementation for SNMP. If network administrators use custom tools to manage their network device inventory they will have to import their data into our tools' schema. With a common representation of the *running-configuration* of a deployed device we can match devices encountered on an audit to pre-measured snapshots in the community database.

B. Using Standard Management Tools for Energy Audits

Network management and operations is a non-standardized, complex and evolving discipline. The uniqueness of infrastructures and diversity between hardware vendors lead to a wide variety of management tools, protocols, and processes. For example, [10] lists hundreds of available commercial and open source tools for network monitoring. Furthermore, many medium to large enterprises and Internet service providers develop proprietary tools to deal with the complexities of their networks.

Important management tasks include tracking inventory and monitoring the status and performance of running network elements. These network elements can span a wide range of platforms such as switches, routers, and appliances such as firewalls and load balancers. This monitoring is required to maintain service guarantees and respond to outages quickly. Examples of these tools include Rancid [5] and commercial tools such as InMon [3] which provide a "single-pane" abstraction that aggregates inventory and flow datasources into a single portal. Similarly, Net-SNMP [2] interacts with deployed network elements via various versions of the SNMPv3 [11]. It provides a set of libraries, visualizations, and network element abstractions that are helpful for administrators for developing their own custom framework. SNMP is a notoriously user hostile management protocol given the need for hierarchical schema descriptions required by a general purpose management protocol. However, Net-SNMP simplifies the task of programmatically interacting with network elements through the SNMP protocol.

All of these tools demonstrate the common requirements for network management. They must be flexible in order to deal with the menagerie of network elements encountered, they must be customized to the requirements of an administrator, and they must export structured reports of the real time system to support configuration modifications that keep the network running within required boundaries.

The first component of EnergyAudit is designed to take advantage of existing capabilities for gathering information about devices, configurations and operational status. A *running-configuration* of a device contains differing levels of detail depending on the data gathering management platform. Minimally, EnergyAudit requires the management platform to collect a fully qualified model name for the device and its components. If a name includes a version number then snapshots between different versions or snapshots without the version number will not report a direct match with entries in our power consumption database (however indirect matching or aliasing can be used when versions are not significantly different as explained later).

Depending on vendor and device, version information can pertain to the software the device is running or differences in the hardware. For a fixed chassis device, ports are components and the port name generally includes the type and position. Modern switches support a "stacking" mode where multiple fixed chassis switches can be combined using proprietary interconnects, EnergyAudit considers these stacked chassis to be components, and the ports contained within the switches are components that have location identifiers pertaining to the particular switch where the port resides.

In a chassis that supports line cards, the line card is a component that requires a fully qualified name similar to a device, while ports are considered separate components. Additionally, each device and component typically have a state (such as up, down, starting). All of these details have a direct impact on the power consumption of the device and are considered by EnergyAudit. Further information such as the type and serial number of the power supplies, switch CPU, switch memory capacity, etc. can also have a meaningful impact on energy consumption but are not currently considered in our methodology. This is a balance between device details that are routinely gathered, and details that provide the best possible match between the device under test and the one stored in the benchmark database.

C. Benchmarking Power Consumption

While a high quality multimeter and power clamp can be purchased for a few hundred dollars, measuring network wide power consumption is a daunting task often ignored by administrators. Network-wide measurement requires access to each device or circuit which can be potentially disruptive for device configurations without redundant power supplies. Further, conducting measurements with multimeters is time consuming across a large campus, and is often not prioritized when network devices are scattered through buildings. Our assertion is that by lowering the overhead and improving the accuracy of gathering measurements of network-wide power consumption more administrators will consider power in network design and management. EnergyAudit is focused on giving administrators the ability to quantify the cost and potential cost savings of various network configurations.

Manufacturers typically report power consumption figures in networking hardware specification documents. However, these values are always for peak power consumption and therefore are only meaningful as an upper bound if a device is operational. Power consumption values are sometimes available in management information bases (MIBs), that are available on devices. However, the details of what is actually being reported in these MIBs varies widely and often does not match empirical measurements made by multimeters [14]. Thus, our approach in EnergyAudit is to develop a database of benchmark power consumption measurements for diverse devices and configurations using standard methodologies (*e.g.,* [23]).

Benchmark measurement of power consumption of networking hardware is the second key component of EnergyAudit. We envision a large database of these measurements based on contributions by device manufacturers, network operators and the research community. Specifically, the database will be composed of multimeter-based power consumption measurements associated with snapshots of a device *running-configuration*. We have already taken over 75 direct measurements of different configurations of 25 commonly deployed devices as a starting point to demonstrate our approach. As the repository grows, the applicability and utility of EnergyAudit will expand. It will be critical to standardize (to the extent possible) on the benchmarking methods that are used so that there is consistency and accuracy in the power consumption estimates produced by EnergyAudit. Manral offers some guidance on power consumption benchmarking in [23].

There are many details in producing meaningful power consumption measurements. Figures 1 and 2 show examples of power consumption measurements of devices made with a high quality multimeter. These measurements were taken with a Fluke 189 multimeter with A/C power clamp. Measurement data points occur at one second intervals and represent the average value of 100 clamp point samples. Figure 1 shows a number of devices in a steady state, where traffic load is relatively constant. In this case it is obvious that measuring and storing the average consumption provides a good match. However, in Figure 2 we see that there can be significant variation in power consumption during a device snapshot that includes the "start up" state, currently to deal with this variation our tool stores multiple snapshots for devices in different states. The stored measurement value for this snapshot remains the average consumption of the device in a particular state. While storing average power consumption is not ideal in every circumstance our system can be easily extended in the future to store vector measurement data or a multi-scalar representation when beneficial.

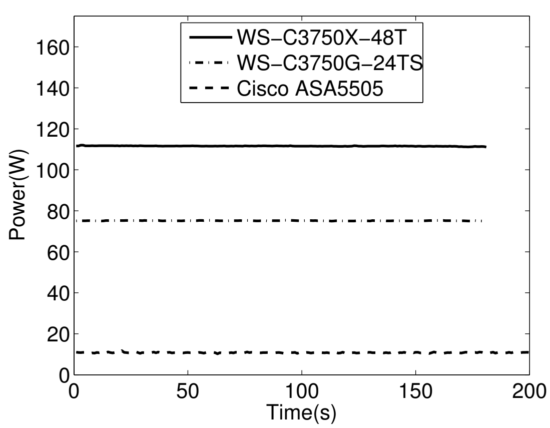

Fig. 1. Measurement of a three different Cisco devices, two switches from different generations, and a firewall appliance (ASA 5505).

Network devices are implemented using a diverse set of hardware platforms. The devices shown in Figures 1 and 2 are fixed chassis switches. They support a fixed number of ports, there are multiple different classes of port 10/100/1000Mbps such as Ethernet and the more modular

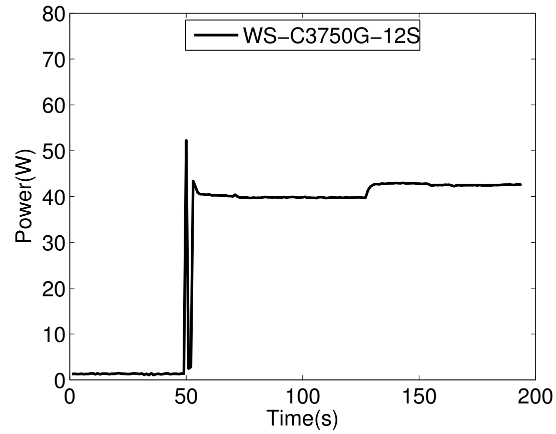

Fig. 2. Measurement of a Cisco 3750-XX during power up.

mini-GBIC ports. Switches can be more complicated, as seen in stacking devices and devices that support PoE, or a relatively simple device such as a desktop switch. Similarly, routers can be expandable based on a multi-slot chassis with multiple cards for route processors and various link media. While we focus on switches and routers in this paper, using a flexible data schema described the Section III we can store direct power consumption measurements with their associated device snapshots for devices such as firewalls and VoIP switches. We currently have two commonly used appliances in the database and expect further inclusion of benchmark measurements of diverse devices in our power consumption database over time.

D. Device Consumption Inference

Directly measuring the power consumption of every potential configuration of every network element is impractical. While there is a non-trivial variety of fixed and multi-slot chassis platforms for switches and routers, enumerating every state for every component would lead to a drastic increase in the measurement overhead. For example, ideally for a given switch we would toggle every port one at a time to capture the full range of power consumption. This is tedious, potentially unnecessary and we would like to simplify the process of adding measurements to the community database. Ideally, every line card in a router is individually measured one at a time, removing those that are not being measured to isolate power consumption. While we try to do this, in some cases it is not possible to bring down a line card as these devices can be deployed in active production networks.

In Figure 3 we show an example direct measurement of two different fixed chassis switch types. For each measurement point, we average at least 1 minute worth of stable power consumption measurements conducted every tenth of a second. We have enabled a variable number of ports for the devices and can see the increasing power consumption trend. We calculate the per port cost for the TS-S model is roughly 1.7W while the per-port consumption of the X-

48T model is roughly 0.45W from this set of measurements. In Figure 4 we explore the relative costs of fixed chassis switches that share a basic model type but differ primarily on the number of fixed ports supported. We can see that the per-fixed-port costs of the base power of these chassis types varies at different rates between models.

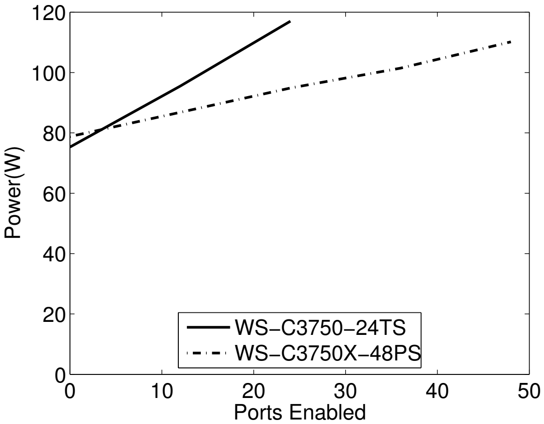

Fig. 3. Measurement of two different cisco fixed chassis switches where we enable 0,12,and 24 ports for the TS-S model and 0,12,24,36,48 ports with the X-48T model. Ports are enabled administratively and then ethernet cables are looped between active ports –activating the port and providing traffic.

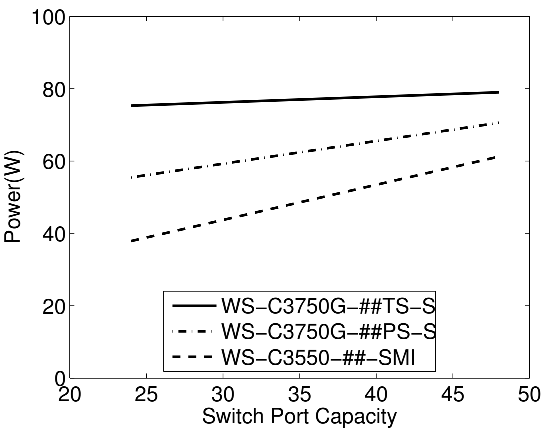

Fig. 4. Measurement of three different fixed chassis switches where the only difference in the model is the number of supported ports. Measurements taken from devices with 24, 48 ports.

Given the trends shown in Figures 3 and 4 we include indirect inference capabilities in our tool. First, we perform *port interpolation* for fixed chassis switches and for line cards. If we have multiple snapshots which are identical except for the number of active ports we calculate the line that bisects the closest two snapshots to the device under test and linearly interpolate the inferred power consumption. Next, if we have switches that have the same hardware but different software versions running we consider them to have the same power consumption, we call this *version inference*. Finally, if there is a device under test where we have a snapshot of the same device type but with different components we take the minimal number of common components with devices with the same name present in the database and report this aggregate power consumption of the minimal match as a *lower bound* inference. While these indirect inference techniques are less accurate than a direct inference, they greatly expand the coverage of our tool and can be disabled by administrators if they are not desired. At this time, we leave indirect inference between device types to future work.

III. EnergyAudit Implementation

In this section, we describe the implementation of EnergyAudit, focusing on the three main components: the management utility API, the community database and auditing. We provide details on requirements, design and development choices and current status for each of the three primary components of the system.

A. *Interfacing with Network Management Utilities*

The first requirement for EnergyAudit is the ability to gather configuration and status information from network systems in a query-response fashion. Recognizing the diversity of network management tools (including commercial, open source and home-spun) in use in networks today, it is obvious that it was neither desirable nor prudent to "reinvent the wheel" when it came to data gathering. Furthermore, it is clear that interfacing with existing tools will increase the potential for uptake and impact by EnergyAudit.

While the design choice of interfacing with existing management platforms has clear and compelling advantages, there are also several challenges, Specifically, there are a large number of network management tools in use today and each one provides unique measurement data which we use as input to our tool. Our goal is not to interface with each and every utility – that would be impractical. Rather, we focus on developing mechanisms that simplify integration and two reference implementations that can serve a guides for future integrations.

EnergyAudit's integration with existing management tools is based on a custom but flexible data representation that is used to describe a network device. Specifically, EnergyAudit translates a device description provided by a given management tool into our common representation using our simple component-based API. The data representation is based on a device and component model. Each device is stored in an object that has a set of identifiers and each device component contains a set of attributes that form a description and a link to the containing device (specific attributes are described in the next section). The translation is done with a customized parsing utility which reads stored management files in the case of an infrastructure management utility, and maintains a hash of OID's to common interface names for SNMP. A number of modular parsing elements are currently implemented in the tool, such as a block parser for switch configuration files scraped from the Cisco IOS command line. This parser goes through the file one line at a time and reads in single command blocks (denoted with a special character such as “!”) that contain recognized output.

The EnergyAudit API provides direct access to the objects that store the device and component representations. Our API is aligned with our community database schema which is described in the next section. Our API supports standard get and set methods for the device name, manufacturer, and type as defined in the data schema. Additionally, each component is required to have a name and a containing device. Additional specific component attributes are listed in the next section.

The first reference interface is with a widely used router configuration manager, “Really Awesome New Cisco confIg Differ” (Rancid). This utility takes administrator provided text input that stores the address and type of device of interest. For each probing round, the tool logs into each network device in the file, runs a customizable set of commands, scrapes the CLI output, and stores the resulting information in a CVS repository. The tool will also send emails to administrators containing the "diff’" between consecutive measurement cycles. Rancid contains a repository of custom scripts that cover multiple devices from multiple vendors. The purpose of the scripts is to parse the various CLI output formats and remove “transitory” values that are not important and would normally change between measurement cycles (such as system time). Our interface to Rancid traverses the CVS repository, parses the stored configuration files, and imports representative device snapshots into EnergyAudit.

The second reference interface is an import utility using SNMP object identifiers. This capability is not a complete implementation due to the complexity and diversity of the SNMP MIB specification. However, it is suitable for interfacing with the management utilities in our test deployments. We envision that as this tool is adopted, the number of SNMP devices and data fields supported will increase.

Administrators who would like to use EnergyAudit with infrastructure management tools other than Rancid or a SNMP variant will have to implement an import method using the EnergyAudit API. Due to the fact that interface, component, and device names are not currently standardized, we implemented an aliasing system. We use aliases internally to match field names which vary on different management platforms. We do not argue that this is an ideal approach. However, we do argue that it is practical and feasible since network management is an evolving area which requires flexibility to accommodate legacy devices.

B. Community Database

The second requirement for EnergyAudit is a database of benchmark power consumption measurements for network devices. The database itself has no complex requirements other than extensibility, scalability and robustness, and is relatively simple. We do not envision it being extremely large (megabytes, not gigabytes) in the short term. The challenge is in populating the repository with consistent benchmark measurements that capture a range of power consumption levels that reflect devices and operating conditions that are commonly found in networks. The higher the fidelity of the benchmark measurements, the higher the fidelity in the resulting aggregate energy consumption reports from EnergyAudit.

The schema for our database is designed to represent a wide range of devices types. Each device snapshot has a manufacturer, device name, and a collection of components. For example, a router has a device name *e.g.,* Cisco 6506 and a collection of components including a chassis and 6 slots that can be filled with various administrative and media cards *e.g.,* Supervisor, 16 port 1000 BaseT-GE, 16 port GBIC, etc. and route processors such as a 4 port 10Gb line card. The CISCO WS-C3750X-48T-SV01 switch as another example, this device has a fixed number of interfaces that report a status such as administratively up or down. "Stackable" devices (*i.e.,* devices that have fixed hardware configurations) are represented as a unique device and collection of components.

Each component has a number of features that we describe below. Some of these fields contain overlapping information in the examples provided above *e.g.,* Type, Name, Description. However, different vendors use different descriptive names for each of these fields. If a particular component type does not support a feature it is marked null. We enumerate the features of the component class with an example field values.

- **Type**: Media type for the component *e.g.,* 1G Ethernet line card,
- **Name**: Specific serial for the component *e.g.,* WS-6516-GE-TX,
- **Description**: Fully qualified text description of component *e.g.,* 16-Port Gigabit Ethernet Switching Module,
- **Version**: Some components have multiple hardware or software versions,
- **State**: Such as up, down, or startup. This field can be customized to include run time states including device load,
- **Ports**: The number of ports on a component. Complex ports can also be stored as individual components,
- **Position**: The slot position of a line card in a router, the location of a port on a standalone switch or chassis line card, or the switch number of a stacked switch.

When auditing network-wide power consumption we attempt to match a candidate device snapshot with one that is recorded in the database. We identify a direct match if the device manufacturer and name are the same and we have components in the database with identical feature values for the **name**, **version**, **ports**, and **state**. We disregard component **type**, **description**, and **position** in a direct match. In the case where direct matches are not found in the community database we attempt to infer the power consumption of the device via *port interpolation*, *version inference*, or *lower bound* inference match as described in Section II-D.

We implement the EnergyAudit community database using MySQL. We have implemented the database bindings with the Python MySQL connector. Specifically, we have device and component tables with the features described above. Additionally, devices are given database unique identifiers and these identifiers are recorded in component records which are added with a specific device. We can use this to quickly pull up available components for a device class for quick matching and inference. Additionally, we implement a user table which links device snapshots and power consumption measurements to a profile for the submitting researcher or network administrator.

We currently have measurements for 25 different devices with roughly 75 different configurations. We will increase this count as the tool is adopted and hope for community contributions. While this number is small compared to the total number of network devices on the market, we have measured a number of extremely common devices and will continue to expand this valuable community resource.

C. Network-wide Energy Auditing

A primary requirement of the EnergyAudit tool is to report accurately on network-wide power consumption. While auditing power consumption, EnergyAudit can use either a central repository of management files (as implemented by Rancid), or pull device details from the running configuration of a set of active devices (using SNMP-walk).

After gathering device snapshots and translating them into the common representation, the community database is queried in an attempt at mapping snapshots to available measurements. In the case of a direct match in the database, the tool records the previously measured power consumption value for that particular device. However, in large diverse deployments and while the community database is lightly populated it is expected that there will not be an exact match present in the database. In this case, depending on administrator preferences the tool can either skip over the device, or attempt an indirect match.

In the case of an indirect match, the tool provides three capabilities. *Port interpolation* is done in the case where there are multiple snapshots that differ in the number of ports with an *on* status. A linear interpolation is conducted between the two closest snapshots. *Version inference* is a manual association between models which are functionally very similar but have a different explicit name. *Lower bound inference* is used primarily with routers and reports the base level steady state consumption for a device when the exact line cards are not present in the database.

After all devices are matched or found to be missing from the the database, the EnergyAudit will generate an the overall network-wide power consumption report for the current monitoring cycle. This network-wide snapshot is stored for for comparison against future measurements.

Device Type	Component	Cost(W)
ASA5580-40 - a PIX	Entire Chassis	503.6
WS-C6509-V-E	Total Cost	1019.7
WS-C6509-V-E	2 WS-X6716-10GE linecards	-
WS-C6309-V-E	2 Distributed Forwarding Card	-
WS-C6309-V-E	Supervisor Engine 720	-
CISCO2921/K9 router	Total Cost	59.1
3845 - a 3800 router	Total Cost	71.3W
WS-C3750X-48T-S	Chassis+48 ports	115.4

TABLE I
ENERGYAUDIT RESULTS FOR THE RESEARCH FACILITY NETWORK DEPLOYMENT.

IV. ENERGYAUDIT TEST DEPLOYMENTS

To test and assess EnergyAudit, we deployed it in three networks on our campus. The first network is in a new, state-of-the-art multidisciplinary research facility that supports thousands of users and includes primarily new networking equipment. The second network is in the computer science department, which supports thousands of users and includes a mix of new and legacy equipment from a variety of vendors. The third is a limited deployment in the campus network, which supports tens of thousands of users and includes a wide variety of devices. We present data from single network-wide snapshots for each deployment, getting multiple snapshots to determine network-wide variability is a focus of future work.

A. Research Facility Network Deployment

The first deployment of EnergyAudit is in a network that has a footprint limited to a single building. The networking systems are all relatively new (purchased within the past 2.5 years). The network is configured in a tiered core-distribution-access topology. As such, many of the switches in this deployment there are of the "stackable" variety. Additionally, there are a number of service platforms for VoIP and policy implementation.

EnergyAudit was configured for direct measurement access to many of the devices in this network. In Table I we list example routers, switches, and appliances that are identified and matched by EnergyAudit. We use the Rancid import utility with the EnergyAudit API to import configuration data into the tool.

EnergyAudit computes that the total consumption for this site to be roughly 5,184W. There are 22 devices in this network deployment when considering a stacked switch as one element, if each switch is considered individually there are 41 total devices. We report the power consumption from 21 of these 41 total devices. Note that we used a non-PoE switch as a *lower bound* for switches of the same model but with PoE (WS-C3750X-48T vs. WS-C3750X-48P). There is one device class (WS-C3750E-48P) which comprises 14 of the missing matches, we have not yet been able to schedule a service outage and measure this particular device, but plan to do so as future work.

This deployment demonstrated that even though the building has power metering at the panel level, each panel is used to support loads from many different devices in addition to the networking infrastructure. As a result, the only way to get a snapshot of the power consumption due to network infrastructure is with direct measurement with a multi-meter or with the EnergyAudit tool.

B. Department Network Deployment

The second deployment is a departmental network that also has a footprint limited to a single building. This network has a greater diversity of devices than the research facility deployment due primarily to organic growth over time.

Specifically, this deployment has 103 total network elements arranged in a tiered access and core configuration. There are 18 total different network element types. Two of these devices are routers and the remainder are switches. There are 3 core routers, 1 border router and the remaining switches that provide access between hosts and the core of the network for this particular departmental network.

Configuration data has been collected from the network administrator's Rancid repository. We use the *lower bound* inference technique for the 4 routers. Figure 5 shows that there are 3 different switch device types that are deployed frequently, demonstrating that while the community database has missing network elements it can be filled with the frequently occurring device types to provide coverage. The deployment is predominately Cisco devices, our tool has matched 74 of the 103 devices in the network and reports an energy consumption of 6,597W.

We do not have direct access to the devices in the network, from this deployment we show that the EnergyAudit tool can estimate power consumption without physical access to the network elements

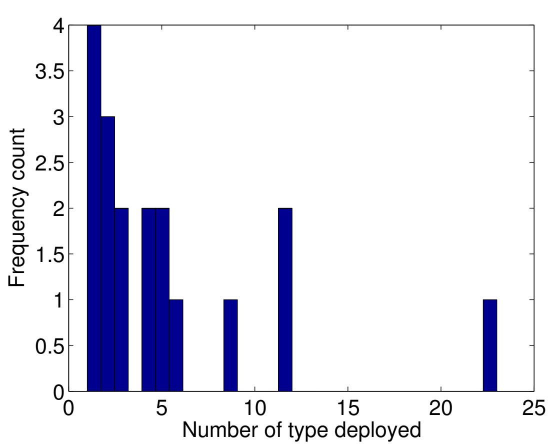

Fig. 5. Histogram of the number of each type of network elements (2 router types, 16 switch types)

C. Campus Network Deployment

The third test deployment for EnergyAudit is in our campus network, which has a metro area geographic footprint. The campus topology is also configured in a tiered core, distribution, access network. For this deployment, we combine two different network management datasources with a customized import utility that uses the EnergyAudit API. We combine a list of SNMP device OID's with the files containing command line output from network elements that support automated command line interface interaction.

There are 5,940 networking devices that are available to the EnergyAudit tool. This is obviously too many devices for convenient direct measurement. We use *lower bound* inference on 12 WS-6506 routers and 10 WS-6509-V-E routers. With the current snapshots in the database, and the *lower bound* inference on the routers EnergyAudit resolves the power consumption of 2,394 devices. These devices are estimated to consume 98,654W. Many of the missing elements, roughly 2,200, are two different models of wireless access points. While we currently don't have snapshots for these access points and they are not included in our audit results, we found that the devices that are powered via PoE reserve 4.4W (an upper bound) from the chassis that provides uplink capability. Figure 6 provides a breakdown of the prevalence of each switch type with a histogram that shows the tool can achieve significant coverage with a small number of additional targeted measurements.

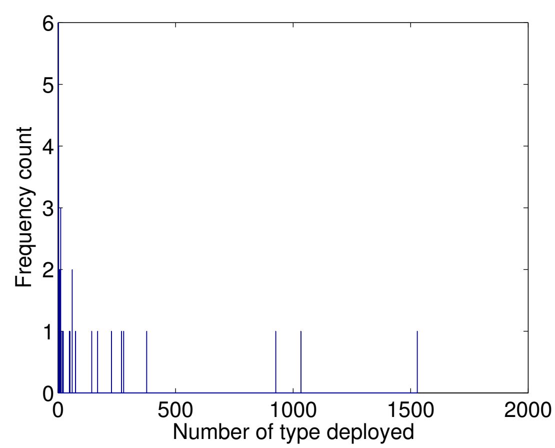

Fig. 6. Histogram of device types for campus-wide network.)

The primary lessons learned from this deployment are that the EnergyAudit tool can provide a power consumption estimate over a large number of network devices quickly, and without extensive administrator interaction. Additionally, with the inclusion of a small number of additional wireless access points coverage could be enhanced. Finally, we demonstrate the SNMP import capability of the tool.

V. Related Work

The energy consumption of networking equipment has been a hot topic over the past several years. In early work, Gupta *et al.* [17] provide a breakdown of the estimated power consumption of networking equipment and note that while the overall power consumption of contemporary switches and routers was small compared to total worldwide energy consumption, switches and routers are *power-inefficient*, which suggests possibilities for optimizations. In the datacenter context, Abts *et al.* [12] argue that while the power consumption of switches is small compared to servers, progress is being made on making servers *power proportional*, while the network power consumption remains relatively static (always on at full bandwidth).

There have been a number of studies that measure the power consumption of individual network devices including [14], [20], [22]. However, these studies do not focus on auditing energy consumption in individual network deployments. Additionally, these studies were focused on measuring specific routers and switches for consumption trends to parametrize a general device model. Our work with EnergyAudit focuses on providing power consumption estimates for network administrators based on the current state of devices in an infrastructure.

With the realization that the power consumption of network elements is due partially to inefficient network platforms and deployments, there have been recent proposals to decrease the power consumption of individual devices and topologies. In [24] Nedevschi *et al.* evaluate the performance envelop of hypothetical device rate reduction versus device sleeping. In their work on shadow ports Ananthanarayanan and Katz propose a novel technique for powering down individual ports while retaining device connectivity [13]. Vishwanath *et al.* propose dynamically adjusting the size of router buffers as they consume non-trivial amounts of power and are not consistently in use [27]. Additionally, new network devices such as those developed by Trendnet [1] are turning off unneeded Ethernet ports when a connected device is turned off or enters power saving mode and modifies port voltage based on Ethernet cable length. Finally, the IEEE 802.3az standard [7] offers a common mechanism for transitioning between PHY's of various speeds depending on link load. All of these capabilities should help to reduce the power consumption in networks but somewhat complicate the task of power auditing due to the different possible states of devices.

At the PoP-level Chabarek *et al.* propose a network-wide optimization of device deployment [14]. Similarly, ElasticTree is a set of optimization and heuristic based techniques for switching off unused sub-trees in a Fat tree datacenter deployment [18]. Additionally, Vasic *et al.* propose REsPoNseTE a system that routes traffic over energy critical paths during periods of low network use and dynamically powers up components of the network as load increases [15] . In [21], Mahadevan *et al.* propose a set of power saving port and traffic consolidation techniques to save power. These device and network-level techniques for reducing power consumption are helpful for reducing overall network power consumption. The role of EnergyAudit is to provide a community utility which eases the task of understanding how much power a network consumes. Additionally, EnergyAudit is built as a flexible framework that can be used by a variety of network management platforms, and EnergyAudit is the first platform to offer a public database of energy consumption of network elements.

Power benchmarking frameworks for networking devices based on direct measurement are described in [23], [26]. Also, the Energy Star program is developing a draft test procedure and efficiency specification for small and large networking equipment [4], [6]. We agree that standardized benchmarking methods are important. Lanzisera et. al. estimated total U.S. IP network equipment energy use in [16]. EnergyAudit includes a direct measurement component to fill the community database and offers additional capability to infer network-wide power consumption directly from device snapshots. Additionally, the IETF/EMAN working group [8] is proposing standards for networking device reporting capabilities including a power and energy MIB. This is ongoing work with limited adoption, EnergyAudit uses direct measurements for matching and works well with legacy and low cost devices which do not have the ability to measure and self-report energy usage.

Phillips *et al.* demonstrate a regression approach for inferring the power consumption of traditional telephone switches [25]. While EnergyAudit can use interpolation to match similar device snapshots, site-wide network power consumption cannot easily be isolated in many contexts and there are a large number of network element types, making a direct regression inference of network element power consumption infeasible at this time. Finally, with their Joulemeter virtual machine power estimation, Kansal *et al.* propose a measurement based power model for inferring the power consumption of virtual machines from device resource usage [19]. Our approach is similar in attempting to use measurement based system snapshots to infer power consumption, while our domain (network elements) is different also we attempt to create a data schema and database of network device snapshots and power consumption as a community resource.

VI. Conclusion

In this paper we describe EnergyAudit, a tool that reports power consumption in a network. Instead of requiring extensive and potentially disruptive direct measurement, EnergyAudit maps device snapshots to a priori power measurements. Existing network measurement frameworks are necessarily flexible to accommodate a variety of network elements and reporting capabilities. As a result, we have designed our tool to work with a common network infrastructure monitor, Rancid and the management protocol SNMP. Additionally, the tool presents a simple API to allow administrators using other custom tools to use the EnergyAudit data schema.

To distribute device snapshots and power measurements, we develop a custom data schema and community database.

The custom data schema ties individual components to network elements and supports the measurement of amortized components and aggregate measurements of multiple components. We have instantiated the community database with over 75 direct measurements from 25 common device types. To our knowledge, this is the largest collection of direct power measurements for switches and routers that is publicly available. The community database and EnergyAudit tool are openly available from www.cs.wisc.edu.com/~jpchaba/energyaudit. Our on-going objective is to expand the corpus of measured device configurations and providing a simple auditing utility that will help elevate awareness of network power-consumption.

We demonstrate our tool on three test networks, a research facility with relatively homogeneous network devices, a departmental network with more diverse network devices and a campus-wide deployment with a large number of devices. In the research facility deployment, our tool provides device granularity power auditing in areas where infrastructure is shared between networking and non-networking equipment. Additionally, we take direct measurements of a number of switches in this building to fill our community database. In the department network, we support a power consumption audit of a medium scale network where we do not have direct access to the devices and due to the number of devices direct measurement would be highly time consuming. Finally, in the campus deployment, we demonstrate the EnergyAudit tool with a network of roughly 6,000 devices. This is a scale that is well beyond what would be feasible for multimeters and shows how EnergyAudit might be used in medium-scale to large-scale networks.

There are limitations in the current generation EnergyAudit tool that inspire future work. First, there are a number of device types which are not present in the community database. While we currently support a number of intra-device class inference techniques, EnergyAudit can use detailed component information that is not universally exposed by network monitoring tools with an inference algorithm such as clustering to infer power consumption between device types. Finally, we seek to extend the EnergyAudit tool to other applications where device snapshots can be associated with a priori measurements in real time such as servers in a datacenter.

VII. Availability

The EnergyAudit tool and database is available upon publication at:

http://www.cs.wisc.edu.com/~jpchaba/energyaudit

Acknowledgment

This work was supported in part by NSF grants CNS-0831427, CNS-0905186, ARL/ARO grant W911NF1110227 and the DHS PREDICT Project. Any opinions, findings, conclusions or other recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF, ARO or DHS.

References

[1] TRENDnet switch. http://www.trendnet.com/?todo=home.
[2] Net-SNMP. www.net-snmp.org, 2002.
[3] InMon. www.inmon.com, 2012.
[4] Large Network Equipment Specification v1.0. https://www.energystar.gov/products/specs/node/413, 2012.
[5] Rancid. www.shrubbery.net/rancid, 2012.
[6] Small Network Equipment Specification v1.0. https://energystar.gov/products/specs/small_network_equipment_specification_version_1_0_pd, 2012.
[7] Energy Efficient Ethernet. http://ieee802.org/3/az/index.html, 2013.
[8] Energy Management. http://datatracker.ietf.org/wg/eman/charter/, 2013.
[9] Net-SNMP. www.net-snmp.org, 2013.
[10] SLAC. http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html, 2013.
[11] SNMP RFC 3410. http://www.ietf.org/rfc/rfc3410.txt?number=3410, 2013.
[12] Dennis Abts, Mike Marty, Philip Wells, Peter Klausler, and Hong Liu. Energy Proportional Datacenter Networks. In *Proc. Intl. Symp. on Computer Architecture (ISCA)*, 2010.
[13] G. Ananthanarayanan and R. Katz. Greening the Switch. In *HotPower*, December 2008.
[14] Joseph Chabarek, Joel Sommers, Paul Barford, Cristian Estan, David Tsiang, and Steve Wright. Power awareness in network design and routing. In *Proc. Of INFOCOM*, April 2008.
[15] Nedelijko Vasic et. al. Identifying and Using Energy-Critical Paths. In *ACM CoNEXT*, Dec 2011.
[16] S. Lanzisera et. al. Data network equipment energy use and savings potential in buildings. In *Energy Efficiency, Volume 5, Number 2*, 2012.
[17] M. Gupta and S. Singh. Greening of the Internet. In *SIGCOMM*, Karlsruhe, Germany, August 2003.
[18] Brandon Heller, Srini Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Sujata Banerjee, and Nick McKeown. ElasticTree: Saving Energy in data Center Networks. In *NSDI*, April 2010.
[19] Aman Kansal, Feng Zhao, Jie Liu, Nupur Kothari, and Arka A. Bhattacharya. Virtual machine power metering and provisioning. In *SoCC*, June 2010.
[20] Priya Mahadevan, Sujata Banerjee, and Puneet Sharma. Energy Proportionality of an Enterprise Network. In *Proc. First ACM SIGCOMM Workshop on Green Networking*, Aug 2010.
[21] Priya Mahadevan, Puneet Sharma, Sujata Banerjee, and Parthasarathy Ranganathan. A Power Benchmarking Framework for Network Devices. In *Proc. IFIP Networking*, May 2009.
[22] Priya Mahadevan, Puneet Sharma, Sujata Banerjee, and Parthasarathy Ranganathan. Energy Aware Network Operations. In *Global Internet Symposium*, May 2009.
[23] V. Manral. Benchmarking Power Usage of Networking Devices. http://tools.ietf.org/html/draft-manral-bmwg-power-usage-02.
[24] Sergiu Nedevschi, Lucian Popa, Gianluca Iannaccone, Sylvia Ratnasamy, and David Wetherall. Reducing Network Energy Consumption via Rate-Adaptation and Sleeping. In *Proc. Of NSDI*, April 2008.
[25] Steven Phillips, Sheryl Woodward, Mark Feuer, and Peter Magill. A regression approach to infer electricity consumption of legacy telecom equipment. In *Proc. Of GreenMetrics*, June 2010.
[26] Ramya Raghavendra, Parthasarathy Ranganathan, Vanish Talwar, Zhikui Wang, and Xiaoyun Zhu. No Power Struggles: A Unified Multi-level Power Management Architecture for the Data Center. In *Proc. Of ASPLOS*, March 2008.
[27] Arun Vishwanath, Vijay Sivaraman, Zhi Zhao, Craig Russel, and Marina Thottan. Adapting Router Buffers for Energy Efficiency. In *Proc. Of ACM CoNext*, December 2011.