Identifying Energy Waste through Dense Power Sensing and Utilization Monitoring

Maria Kazandjieva, Omprakash Gnawali, Brandon Heller, Philip Levis, and Christos Kozyrakis
Computer Science Lab, Stanford University
mariakaz,gnawali,brandonh,pal@cs.stanford.edu, christos@ee.stanford.edu

Abstract

PowerNet is a hybrid sensor network for monitoring the power and utilization of computing systems in a large academic building. PowerNet comprises approximately 140 single-plug wired and wireless hardware power meters and 23 software sensors that monitor PCs, laptops, network switches, servers, LCD screens, and other office equipment. PowerNet has been operational for 14 months, and the wireless meters for three months.

This dense, long-term monitoring allows us to extrapolate the energy consumption breakdown of the whole building. Using our measurements together with device inventory we find that approximately 56% of the total building energy budget goes toward computing systems, at a cost of $\approx \$22,000$ per month. PowerNet's measurements of CPU activity and network traffic reveal that a large fraction of this power is wasted and shows where there are savings opportunities.

In addition to these sensor data results, we present our experiences designing, deploying, and maintaining PowerNet. We include a longterm characterization of CTP, the standard TinyOS collection protocol.

The paper concludes with a discussion of possible alternatives to computing system design that can save energy while satisfying user workloads.

1 Introduction

This paper describes the design and deployment of PowerNet, the first sensor network that monitors both power draw and utilization of computing devices in an office building. PowerNet provides fine-grained data on approximately 140 devices, including many different desktops, monitors, network switches, and servers. Two types of power meters, wired and wireless, collect measurements once a second, while software sensors gather statistics on computer and network utilization. PowerNet has been active for 14 months, with the most recent deployment phase completed in January.

To date, "green computing" has focused primarily on the data center, as it represents a large cost that lends well to centralized control and optimization. This paper takes a different approach, examining a computing-heavy office building. Compared to a data center, this environment has a tremendously diverse set of devices that exhibit huge variations in workload and configuration and exist under several overlapping administrative domains. Improving the efficiency of such a computing system requires detailed data of both energy consumption and energy waste.

This paper makes four contributions. First, we extrapolate a whole-building breakdown of energy consumption by combining PowerNet data with surveys, observations, and IT database records. We estimate that at least 56% of the building's \$40,000 monthly electricity bill goes to its computing systems infrastructure.[1] Desktops and laptops consume 16%, 30% goes to servers, 7% goes to monitors, and 3% of energy goes to networking.

Second, we quantify power variation between device classes, within device classes, and for individual devices. This analysis identifies simple optimizations, such as changes in display settings, that lead to significant energy savings.

Third, PowerNet's ability to correlate power consumption with utilization allows us to differentiate energy used well from energy waste. This is an important difference from previous work [8, 14]. We pinpoint examples of energy waste caused by over provisioning, such as the wired building network. We also identify policies that prevent energy conservation, such as a nightly backup policy that requires desktops to be kept on overnight even though backups only take an hour. Finally, we turn the power and utilization data into insights for future computing infrastructure decisions: purchasing high-end vs thin-client desktops and the benefits of intelligent power management.

Moreover, the nature, scale, and duration of the PowerNet deployment yields insights for future indoor sensor deployments. We present the first long-term study of CTP [7], the standard TinyOS data collection protocol. We find daily and weekly cycles in CTP's behavior, caused by human activity, and present a crippling bug whose fix has since been included in the standard TinyOS release. Our experiences deploying PowerNet will inform future efforts to understand and reduce energy consumption in office-style buildings.

2 Overview

The PowerNet deployment collects power and utilization data for the computing infrastructure of an office building. Figure 1 shows the overall design. The deployment currently

[1] Cooling and heating are through chilled water and steam, so are not part of this total.

Device Type	Count	Utilization	Count
Desktop	44	CPU	15
Monitor	40		
Laptop	16	CPU	1
Network Switch	11	traffic	7
Printer	10		
Server	9		
Fridge	3		
Access Point	2		
External Hard Drive	2		
Fax Machine	1		
Total:	**138**		**23**

Table 1. PowerNet covers a variety of devices whose power measurements we use to characterize the energy consumption of the whole building. We also monitor CPU utilization and network traffic.

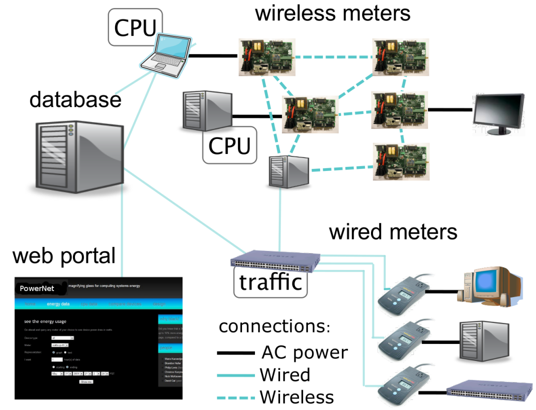

Figure 1. The deployment measures power usage and utilization of individual devices. The data is transmitted over the network and stored on a central server.

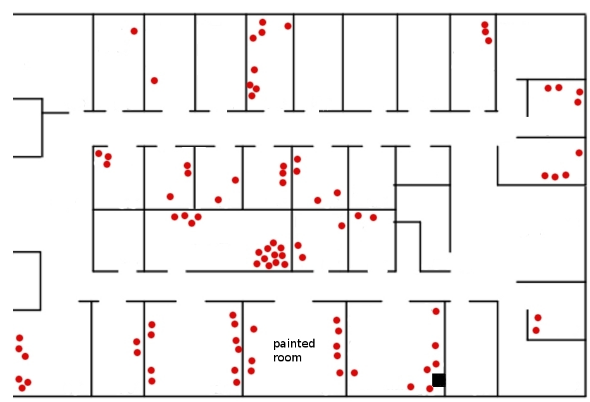

Figure 2. The wireless power meter deployment spans one wing on one floor of an office building. The black square represents the sink and every dot is a power meter. Most meters are located under desks, near the floor.

includes 138 single-outlet, high-sampling-rate power meters, both wired and wireless. Each meter connects to exactly one computing device, such as a PC, display, or network switch; the full list is shown in Table 1. In addition, building occupants and system administrators have volunteered to install software to monitor CPU utilization and network traffic. All data is logged continuously to a central MySQL database, from once a second to once a minute, depending on the source. The system currently logs approximately 1 GB per day from 161 data sources.

Wired Meters. The 55-node wired deployment is sparse, covering spread-apart wiring closets, student offices, and a basement server room. Commercially-available Watts Up .NET meters [10] transmit measurements over Ethernet, up to once a second, over the existing building network. Each meter posts data via HTTP to a server process on the PowerNet logging machine. These meters were a useful first step in gathering power data, though the practical issues of size and proprietary software (described later in Section 5.1) hindered further deployment.

Wireless Meters. In contrast, the 85-node wireless deployment is dense, covering a large fraction of the power outlets in one wing on one floor, shown in Figure 2. Custom-made low-power wireless meters transmit data from an ad-hoc multihop network. Each meter is a modified version of the open-source ACme meter [14]. Each ACme includes power measurement circuitry and an Epic core with microprocessor and radio chip [11]. More hardware details can be found in Section 5.2.

The meter software, built on TinyOS, includes sampling, routing and dissemination capabilities. The top-level application reads power draw every second and sends a data packet after buffering ten samples. The motes use CTP [7] as the underlying routing layer. The code includes Deluge [13] for remote image upgrades. In addition to power data, the motes gather CTP statistics. Section 6 describes in detail the gathered data and the resulting observations. To our knowledge, this deployment is the largest, longest-term, and highest-density one using CTP, and one of the first to be done indoors.

Utilization Monitoring. PowerNet also monitors utilization, in addition to power, for 23 computing devices. On the network side, an SNMP script polls seven network switches once a minute and records the average traffic in Mbps. On the PC side, we collect CPU utilization and the list of active processes. Seven student PCs run a script that reads the /proc virtual file system to give average CPU load every second. Nine staff machines run a Windows script that reports the list of running processes (similar to the task manager) and total CPU load. The combined data helps correlate power with observed workload.

Data Storage The wired and wireless meters and CPU monitors send the data to a central server with two 1.8 Ghz cores and 2 GB of RAM. With over 160 sensors reporting as often as once a second, data piles up quickly. Section 5.3 describes the backend scalability challenges.

Data Access and Analysis The data stream provides near-real-time feedback to building residents, equipment purchasers, and system administrators through the PowerNet website. A display in the building lobby provides information about the project, along with graphs showing real-time power consumption of categories of devices, such as monitors, servers, and network equipment.

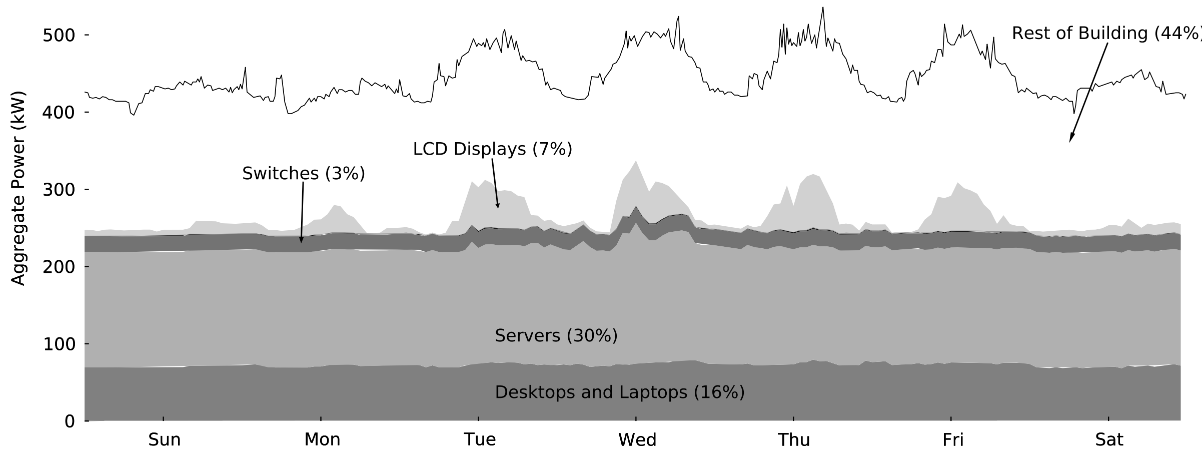

Figure 4. Aggregate power draw for the entire PowerNet building shows diurnal and weekday/weekend patterns. Computing systems account for 56% of the total 445 kW. The given week of data is representative of the building, except Monday which was a university holiday (Feb 15).

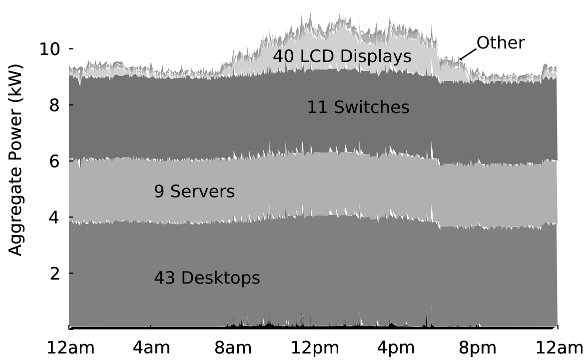

Figure 3. PowerNet's measurements account for 2.5% of the building's power consumption.

However, power feedback for building residents is not the focus of this paper. Our goal is to answer building-level questions about where energy is going and how we change management processes and purchasing decisions to reduce energy consumption. At the heart of this paper are the insights revealed by over 150 gigabytes of collected and correlated power and utilization data. What follows are our findings – expected in some cases, and surprising in others.

3 Computing Energy Consumption

This section analyzes the power data that PowerNet has collected. It examines three classes of devices in detail: displays, computers, and networking switches. Before PowerNet, the building manager's only view into energy consumption was a monthly electricity bill of approximately $40,000. Using PowerNet's measurements, network activity logs, a survey of building occupants, and cross-correlating with IT databases, we find that computing systems draw on average 252kW: 56% of the building's 445kW. We find that displays are responsible for 50% of the building's diurnal power draw variations. There are significant activity and power profile variations within a device class, such that dense sampling is necessary for accuracy: for example, sampling only 5 random desktops has a 5th percentile error of underestimating by 31% and a 95th percentile error of overestimating by 40%.

3.1 The Big Picture

Figure 3 shows the power draw of PowerNet's 140 devices over a 24-hour period. The 9 to 11kW draw is 2.5% of the building total. Each layer represents a different device category. The largest contributors are labeled. Consumption of most devices is fairly steady, with the exception of displays. Although the measured displays are responsible for a 14% increase in the measured power draw, effective power management reduces their total energy contribution to only 7%. Weekends see power profiles similar to Figure 3, although variations are smaller because fewer displays are turned on. Overall, displays are almost completely responsible for the variation we see in computing power.

Even though PowerNet only measures 2.5% of the building power, by combining these samples with other information about the distribution of computing devices, we can extrapolate to the whole building, as in Figure 4. Specifically, from network administrator databases of active nodes, email surveys, and manual inspections of networking closets and server rooms, we can generate a reasonable inventory of devices in the building. We can use PowerNet's measurements to couple this inventory to power draw.

Using PowerNet as a motivation, we convinced campus services to provide Excel spreadsheets of the building's average draw over 15-minute intervals. The top curve in Figure 4 shows one week of this data. Finer-resolution data makes it easy to spot expected trends such as day/night and weekday/weekend patterns: daytime sees a 30% increase in power

Device Type	Measured	Total	Extrapolated via	Total Draw	Uptime	% of Building
Desktops/Laptops	44	742	whois, MAC address registrations	70 kW	24 hrs/day	15%
Servers	9	500	manual inspection	137 kW	24 hrs/day	30%
LCD Displays	40	750	occupant survey	61 kW	12 hrs/day	7%
Switches	11	62	network admin records	15 kW	24 hrs/day	3%

Table 2. We cross-correlate PowerNet measurements with IT databases to extrapolate energy consumption of computing systems in the whole building.

Type	# Count	Power Draw (watts)
HP 5406zl (6-slot)	20	325
HP 5412zl (12-slot)	8	500
HP 2724	2	100
Cisco Cat 6509	2	400
Cisco Cat 4000	2	600
Cisco Cat 3750G	2	160
Linksys	2	50
NEC (misc)	5	100
Cisco (misc)	5	100
Quanta (4-slot)	5	50
Others	9	50
Total:	**62**	

Table 3. Summary of groups of switches with individual and estimated total power consumption. This inventory includes all major network switches in the PowerNet building.

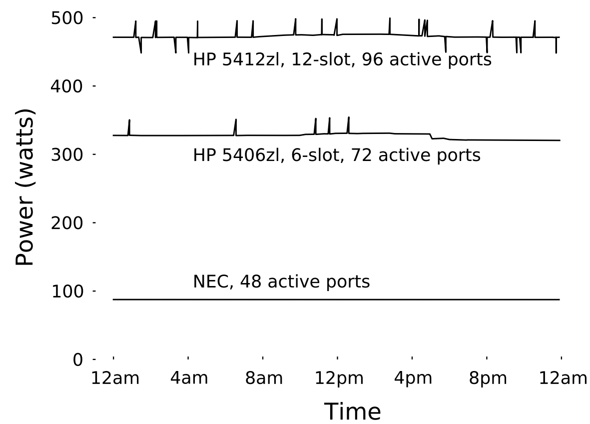

Figure 5. Switch power consumption is constant, barring transient ups or downs likely due to fans. Thus, the networking infrastructure does not require long-term power monitoring.

draw over nighttime. Figure 4 also shows our estimate of computing's contribution to this power draw for the same period. Computing systems are responsible for 56% of the building's total energy consumption. Furthermore, displays are responsible for a 46kW increase in daytime power draw or 50% of the total increase.

The rest of this section examines displays, desktops, and switches in greater detail and explains how we perform this extrapolation. Table 2 shows our extrapolation methodology and results at a glance.

3.2 Networking Equipment

We compute the energy consumption of the building's networking infrastructure (its Ethernet switches and WiFi access points). In the PowerNet building, network access is provided by 2 core switches located in the basement and 26 edge switches spread across the five floors. In addition, there are a number of smaller switches, deployed in ad-hoc ways by individuals or research groups. Table 3 presents an inventory of networking equipment. While we can account for all major switches, finding all of the small ones (e.g., 4-hub Linksys switches) was not possible due the scale of the building and permissions required. Since the infrastructure is planned by a small number of administrators and centrally purchased, the equipment used is comparatively uniform: over half of the switches are a single model, the HP5406zl.

Figure 5 show the power consumption of three switches. Power draw variation is negligible and the small peaks are likely due to CPU spikes. Networking infrastructure power easy to characterize; future deployments need not collect fine-grained long-term samples, since the data rarely changes. We use measurements from the core and edge switches and cross-correlated that to data-sheet-reported values to estimate the total energy consumption. On average, the switches in the building consume 15 kilowatts. This comes to a total of 12000 kWh per month, 3% of the building's total consumption.

3.3 Computer Displays

With the recent shift from CRTs to large LCDs, displays have become a significant contributor to electricity bills. This section examines PowerNet's display power measurements and explains how we extrapolate to the energy consumed by all displays in the building.

Over 600 people use the PowerNet building as office space. PowerNet's power meters and utilization sensors cover a broad and diverse range of residents, including students, professors, visitors, servers, and administrators. While this diversity allows us to see a breadth of usage patterns, it constitutes a highly biased sample. In practice, most of PowerNet's offices are occupied by graduate students. Therefore, simply using a multiplicative factor on PowerNet's measurements could be highly inaccurate. For example, administrators tend to have lightweight desktops and smaller LCDs, while many students have powerful desktops and larger LCD displays.

PowerNet's measurements allow us to quantify the average power draw of a class of display; extrapolating to whole-building power draw requires knowing the distribution of display classes. To obtain a reasonably accurate estimate of

Size	Count	Avg. Power
< 17”	9	
17” to 19”	33	35 W
20” to 22”	40	50 W
23” to 25”	84	66 W
26” to 27”	15	120 W
29” to 30”	42	135 W
> 30	2	

Table 4. A survey shows that majority of building occupants use mid-sized LCD displays. Equipment upgrades cause the number of large (30”) monitors to increase.

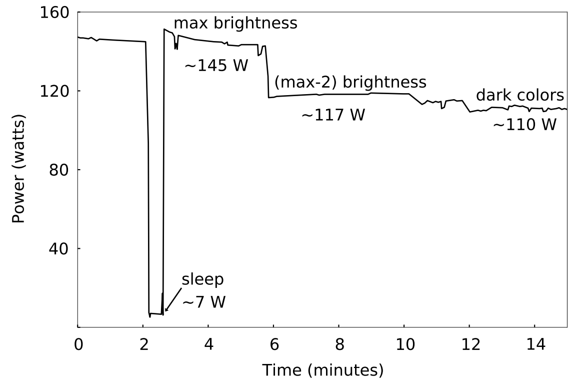

Figure 6. The power consumption of a computer monitor varies widely depending on its settings. Minor adjustments of brightness level can result in 20% savings for large monitors.

this distribution, we distributed an online survey asking occupants for the number, size, and manufacturer of the computer screens they use. Table 4 presents data from the 169 responses reporting 225 monitors and indicating the distribution of sizes. The majority of people use 23- to 25-inch monitors. 30” screens are the second largest population.

Table 4 also shows the power consumption the specification sheets of different displays report. PowerNet’s measurements reveal that there is a great variation in active power draw even between devices of the same size and make. We conducted a controlled test to see how different display settings affect monitor power draw. We chose a 30” Dell monitor, partially to highlight the differences in monitor states, but also because these displays form an increasing portion of the display population.

Figure 6 shows an hour-long data trace during which we adjusted the monitor brightness and desktop color scheme. Depending on the monitor brightness settings and the colors in the image displayed, the power draw varies by up to 35W (25%). Lowering the brightness by two settings (pressing the ’-’ button twice) reduced the average power draw from 145 to 117 watts, a 19% reduction in consumption. Additionally, LCD power draw is affected by the colors displayed. More energy aligns more liquid crystals in each pixel, permitting more light to shine through and enabling them to display brighter colors. Thus, a 30-inch monitor has maximum power draw, measured at 145 watts, when the majority of the screen displays white elements. Switching to a dark background and color scheme or viewing darker web pages reduces the draw to 127 watts. Displaying dark colors with the lower brightness setting reduces power draw to 110W.

With a good understanding of individual power, we can characterize the effect at scale. In order to account for all monitors in the building, we extrapolated from the survey data and typical power draw. From building inventory and network reports of what computers are active, we were able to estimate that the PowerNet building has approximately 750 displays. We assume that these 750 displays follow the same size distribution as gathered in the survey and that they follow the same use distribution as the metered monitors, as there is no significant difference in monitor activity between classes of residents. These calculations indicate that on average, a monitor draws just over 80W, with 750 monitors drawing 61 kW. Since most displays are powered on about 12 hours a day, this translates to 7% of the building total energy.

3.4 Personal Computers

Personal computers – dekstops and laptops – and servers are the largest contributor to the computing infrastructure energy consumption. According to the department’s database of registered devices there are $\approx$1250 machines active on the building’s network. This number includes student, staff, and professor machines as well as server machines located in two server rooms. We manually inventoried the server rooms to distinguish what portion of the 1250 are desktops or laptops and what portion are servers. Of this total, 500 machines are servers, while the rest are laptops or desktops. We refer to desktops and laptops as personal computers (PCs), as many laptops are used with docks.

PowerNet measurements of 44 PCs show that desktops vary greately in power draw – anywhere from 40 to 350 watts. Figure 7 shows the power consumption of three different PCs over 24 hours. Desktop ‘a’ is a Dell Inspiron 530 desktop with a powerful graphics card; desktop ‘b’ custom-built machine and desktop ’c’ is a lightweight Dell Optiplex 745. Power consumption varies widely, not only between desktops, but also for the same desktop in time.

Figure 7 shows that dense, fine-grained, long-term instrumentation is the key to accurately characterizing the power consumption of a building’s computers. To explore this further, we run statistical analyses on the average desktop consumption. The average power draw of the 44 measured desktops is 107 watts. What error could we expect if only 5, 10, or 20 of the desktops were monitored? To estimate the error with only 5 desktops, we generated 1,000,000 random 5-tuples drawing from the lists of 44 desktops. Next, we calculated the mean for each set of 5 machines and plotted a histogram of the results. The experiment was repeated for 10- and 20-tuples of computers.

Figure 8 shows the three resulting histograms with the 44-node mean indicated by a vertical line. As expected, larger sample sizes yield a narrower spread, with averages that are closer to the mean. We calculate the expected error by averaging over the probabilities of all possible mean values as given by the histogram. With 20 desktops , the expected error

	Laptops	Low-end PCs	High-end PCs	Total
observed	47	43	366	**456**
estimated	29	27	230	**286**
Total	**76**	**70**	**596**	**742**

Table 5. Personal computers are binned in three categories, and university databases and active network node counts allow us to extrapolate to the whole building.

in calculating the mean is almost 7% and as much as 17.2% for a sample size of 5 machines. With 5 measurements, the 5th percentile is 31% lower than the 44-node mean, and the 95th percentile is 40% higher.

Such analysis is important when choosing what and how many devices to monitor. For example, in the Green Soda deployment [14] only 10 desktops were measured: Figure 8 shows that such small samples can limit accuracy. Given the diversity of desktops, even denser sampling within this deployment class is our next deployment priority.

To extrapolate to the whole building we bin PCs in three classes – laptops, low-end desktops, and high-end desktops. Low-end desktops are those with average power of about 80 watts or less and include machines such a Mac Minis, Shuttle PCs, Dell Optiplex. Full-size desktops like the Dell Precision are considered high-end machines.

A snapshot of MAC addresses recently seen on the building network returned 1242 active nodes, 500 of which are servers. We took the remaining 742 addresses and cross-referenced them with the university's whois database. This database includes the node description provided upon network registration. Of the 742 nodes, 456 had description that allowed us to classify them as laptops, low- or high-end desktops. The remaining 286 nodes had blank entries in the whois database. Table 5 shows the number of machines in each PC class; nodes with available description are labeled as 'observed' and breakdown of the other 286 assumes that the observed distribution is representative of the building.

We use the 44 desktops and 16 laptops measured by PowerNet to extrapolate power to the whole building. The median power draw for laptops is 25 watts, for low-end machines – 52 watts, and for high-end machines – 108 watts. This data together with the numbers of devices in each bin, given in Table 5 yield an aggregate power draw of 70 kW for the 742 personal computers in the building, or 16% of total energy consumption.

PowerNet server sampling is sparser than with PCs for two reasons. First, servers are densely deployed in machine rooms, which makes it harder to install meters. Second, it is much harder to convince people to allow a meter on critical servers or compute clusters: a meter failure or accident could harm critical data or knock out a large number of devices. A manual inspection of servers and their configurations, however, finds that they have much less variation in power draw. We therefore assume a power draw of 275 watts per server, typical for the rackmount 1U servers in the PowerNet building. With 500 servers, the aggregate draw is 137kW. Together, PCs and servers account for 207 kW, or 46% of the total building power draw.

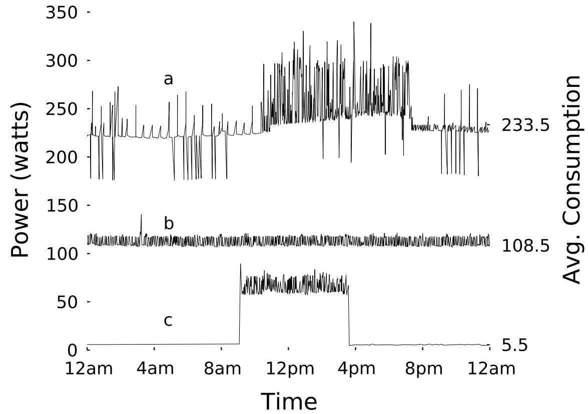

Figure 7. Desktop computers consume a steady amount of energy, but there is great variation between PCs. For example, some staff machines consume only about a quarter of what a graphics student's machine does.

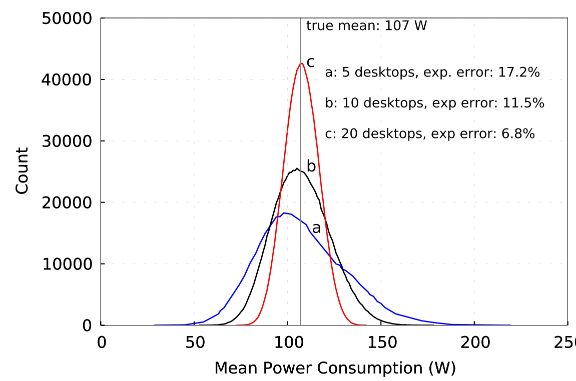

Figure 8. Desktop energy consumption varies over time and device make. Using only 5 desktops to extrapolate to 44 yields an expected error of 17.2%., indicating that dense power monitoring is necessary.

3.5 Summary

PowerNet's dense sensing deployment allows us to extrapolate from 140 meters to aggregate building energy consumption. Overall, we find that 54% of the building's energy goes to computing equipment. 30% goes to servers, 16% to PCs, 7% to displays, and 3% to networking infrastructure. Going beyond how energy is consumed, how much is used for computation and work, and how much is wasted on idle systems? Answering this question requires more than power measurements. It requires a system that measures utilization. The next section describes this second sensing modality.

4 Computing Systems' Utilization

Section 3 characterized the power consumption of an office building. A breakdown of an electric bill is useful because it pinpoints the components that draw the most power, highlighting opportunities for savings. At the same time, it is difficult to say what improvements can be made to the computing infrastructure, if we do not understand the underlying

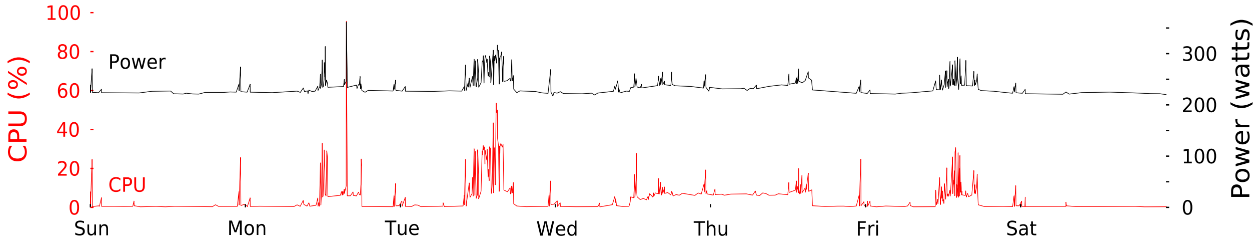

Figure 9. A week-long trace of power consumption and CPU utilization shows idle periods during which the power-hungry desktop could have been turned off.

usage patterns that require computing in the first place. This section digs deeper into the meaning of energy efficiency by correlating power consumption with device utilization.

In an ideal world, all systems would be power proportional, drawing power when work is done, and consuming nothing when the system is idle or unused. Reality is not so kind. We examine the utilization of computers and network switches. The key insight is that current systems, computing or networking, are heavily underutilized. This fact, combined with large baseline power consumption, means that energy efficiency is extremely low. A large portion of the time, electricity bills pay for unused or under-utilized devices.

4.1 CPU Utilization

The aggregate power graphs at the beginning of Section 3 suggest that most computers are rarely turned off. Figure 9 shows power consumption and CPU utilization for one specific computer over 1 week. Usage patterns are immediately obvious: there are long idle periods at night and on weekends. While machine utilization varies greatly over the span of a week, from 0% to 60%, this desktop's draw never drops below 220 watts. Measurements from multiple desktops show an additional cost of roughly one watt for every 1% increase in CPU utilization beyond idle.

If these computers are mostly idle, then why are they not being put to sleep? Going back to Figure 7, only one of the three machines was put to sleep during non-business hours, while the other two remained on. We do not see strong diurnal variation in building power consumption largely because residents are not taking advantage of the sleep and hibernate states provided by modern OSes, especially during nighttime hours.

The reasons for this behavior vary but most often people cite unwillingness to wait for machine startup in the morning, ability to access the machine remotely, and nightly backups. On several accounts, staff members in our department shared that they would love to put their computers to sleep at the end of the workday but are not allowed to do so. Backups are scheduled to begin at 8:45 pm. Backups are one example of a workload that requires a machine to be powered on.

The energy waste from always-on computers is only half the story. Further examination of CPU data shows that even when actively used, most computers are rarely pushed to their processing limits. Table 6 shows the the 5^{th}, 50^{th}, and 95^{th} percentiles of CPU utilization for seven student machines. The data was collected every 1 second for the past 11 months.

Machine Type	Percentile CPU 5^{th}	50^{th}	95^{th}
high-end custom-built	0%	1%	57%
Dell Optiplex 745	1%	9%	58%
Dell Precision T3400	0%	4%	29%
Dell Precision T3400	0%	1%	13%
Dell Inspiron 530	1%	1%	8%
HP Pavilion Elite m9250f	0%	0%	25%
Dell Precision T3400	0%	1%	7%

Table 6. CPU utilization for 7 student machines collected over 11 months reveals high under-utilization.

The measured computers rarely use even 50% of their available CPU.

This observation raises the question of whether powerful desktops are the best way to provide computing power to users. The trends we see are towards upgrading to more powerful machines, yet typical workloads hardly tax the available CPU resources. Section 7 goes further into alternative providing computing systems that meet user needs in a more energy-efficient manner.

4.2 Network Traffic

In Section 3 we found that the networking infrastructure consumes much less energy than desktops. We also noted that switches consume a constant amount of power. This prompts the questions of how much traffic is flowing through the 60 or so switches in the building, and whether that traffic changes with time.

Figure 10 shows the traffic coming into one of the four switches on the second floor of our building. This is an HP Procurve switch with 96 1-gigabit active ports, consuming 500 watts. Over one week in March, bandwidth demand never exceeded 200 Mbps – an amount that could be handled by one gigabit port instead of 96

To verify that this is not aberrant behavior, Figure 11 shows the cumulative distribution of traffic for 7 building switches. Note that the x-axis has a log scale. Table 7 accompanies the figure with a list of switch types we measure and the length of each data trace.

Similar to computers, switches are highly underutilized. For the equipment we measure, total network demand is lower than 1000 Mbps 100% of the time. Of course, network over provisioning is not a new concept or observation; it provides benefits, including higher throughput, lower loss,

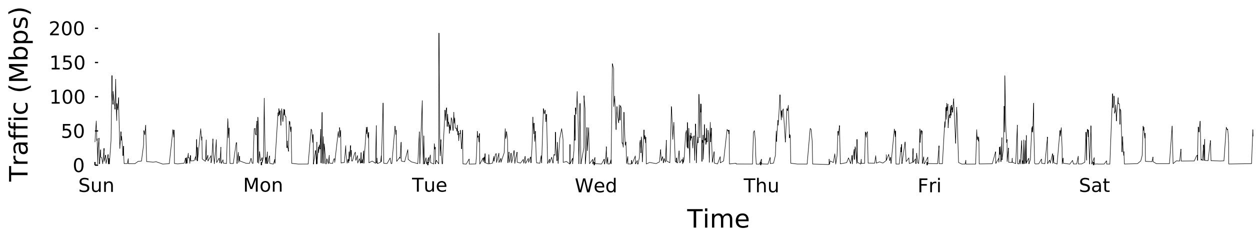

Figure 10. Typical traffic patterns for one edge switches in the building. Network utilization remain low. Power consumption for this switch remain constant, at approximately 500 watts.

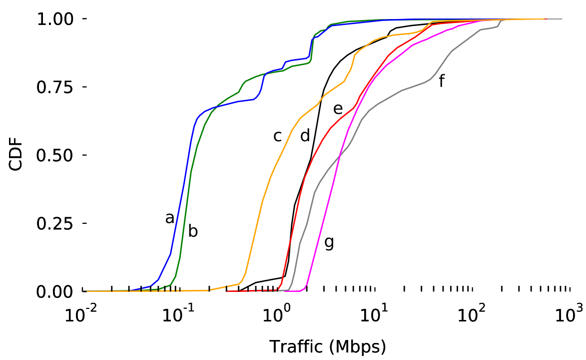

Figure 11. CDF of traffic for seven switches over 6 months shows that switches are operating well under capacity.

Label	**Switch Type**	**Active Ports** (gigabit each)	**Datatrace** (# days)
a	HP 5412zl	120	150
b	HP 5406zl	96	40
c	HP 5412zl	120	40
d	HP 5406zl	72	150
e	NEC IP8800	24	420
f	HP 5412zl	24	420
g	NEC IP8800	48	420

Table 7. Summary of groups of switches with individual and estimated total power consumption. Gates building.

and lower jitter. When the average utilization is under one hundredth of one percent, several questions beg an answer. Is the amount of over-provisioning unnecessarily large? How can we take better advantage of the large amount of bandwidth that today's networks are ready to support? We discuss possible answers to these questions in Section 7.

5 Deployment Experiences

Prior sections presented the data that PowerNet has collected. The next two sections present our experiences deploying PowerNet. This section describes in detail our monitoring infrastructure for collecting power and utilization data. PowerNet uses two types of power meters to collect data; the first are commercial off-the-shelf, while the second are custom-made. We also share experiences and lessons learned over the lifetime of the deployment.

5.1 Wired Deployment

The initial requirement for the power meters was the ability to sense individual outlets at high sampling rates. This differs from many residential solutions that track whole-house energy consumption and report data every 10 or more minutes. Commercially-available Watts Up .NET meters were the first power sensors in the deployment, since they were easy to obtain [10]. These meters transmit measurements over Ethernet, up to once a second. Meters were placed in wiring closets, the basement server room, and spread-apart offices. While these meters were a useful first step in gathering power data, deploying and maintaining them proved to be difficult; problems surfaced even before the deployment began.

The first practical issue was the lack of in-field upgradable firmware. When a bug was discovered in the TCP stack, our only option was to pack up four large boxes of power meters and send them back, so that company staff could fix the proprietary code. After several weeks, the meters were back in our possession and the deployment could begin.

It quickly became clear that few offices had an open Ethernet port for each power meter. Many offices required additional small Ethernet switches and extra cables. The volunteer participants were unhappy with the clutter under their desks, due to the size of the meters. Each one weighs 2.5 lbs, with a thick, six-foot-long cord leading to a 7" x 4" x 2" base. Despite the physically clunky deployment experience, we were able to install 80 meters.

In the PowerNet building, each device must have a MAC address registration to obtain an IP address. Each group within the building has a unique VLAN, and each meter was statically registered to a group. The registrations could not be done all at once, since neighboring offices may correspond to different groups, and we could not know in advance how many meters would be needed for a given office. The network admins were burdened by the power meter registrations, and with this much manual configuration, mistakes happened.

We received an email from a network admin stating that "more than half of all DNS lookups emanating from [the three Engineering buildings] to the campus servers" were coming from the power meters. The solution for the lack of DNS caching was to go back to each meter, plug it into a laptop via USB, and hard-code the IP address of the PowerNet server.

In addition to DNS lookups, the meters were also making ARP requests once per second and overwhelming the network security monitoring infrastructure. We received another email from the IT staff, pointing out that "[t]he 70 current meters now account for 20% of total daily recorded flows" by the security system. To work around this problem, the logging server was moved to a special VLAN that was not monitored by the network admins. That resulted in an IP address change, which meant yet another trip to the individual meters to update the hard coded IP address of the server.

Once the deployment was in place, we observed a number of meter software errors. From the 90 power meters, 8 completely stopped working; they did not power up or did not send or display any data. Another set of 5 to 7 meters began reporting incorrect data at some point of the deployment; from the reported numbers we guess it was an integer overflow issue but the closed firmware did not allow us to verify this. The erroneous data was purged from the analyzed data sets. There were also some meters that would stop reporting data over the network until they were rebooted. That again was likely a software problem where the meters were reverting to logging data locally instead of pushing it out via HTTP. Of the original 90, only 55 are still in operation; a number of residents simply unplugged their meters.

To their credit, the wired meters generally reported accurate data and work well for a dispersed deployment such as the wiring closets. However, three key issues made the wired meters unsuitable for large-scale deployment: the lack of code accessibility and remote firmware upgrade, the overhead of installing the meters within the building network, and user dissatisfaction with clutter and frequent maintenance. These experiences suggest that zero-configuration networks that automatically form distinct subnets (e.g., as is proposed in RPL [6]) would improve ease of deployment.

5.2 Wireless Deployment

Open-source low-power wireless meters were the main candidates for expanding the PowerNet deployment - in particular, the wireless ACme meters used in the Green Soda project [14]. The PowerNet wireless meters are based on the ACme design, with two small modifications. The first was a switch from a solid-state relay to a mechanical one. This change enabled a sealed case, by removing the need to machine side slits to dissipate heat from the solid-state relay. The second change was to add an expansion port with a range of serial interfaces, to support new sensors and added storage. The cost per meter was about $120, as compared to $189 for the wired meters, both in quantities of 100.

The deployment of 85 wireless meters took several afternoons, compared to two weeks for the wired meters. The benefits of the wireless deployment were noticed immediately, and some users even requested that we replace their wired meters with wireless ones. The IT staff was not burdened by meter registrations, and the open nature of the software and hardware made modifications easy. The main meter limitation is transmission distance. Since the PowerNet wireless deployment focuses on a single wing of a building, the range was sufficient for CTP to form a mesh without a need for repeaters.

5.3 Backend and Scalability Challenges

The PowerNet infrastructure currently gathers 1GB of data every day and this number will grow with the next round of utilization sensors and 300 more wireless power meters. When the logging server was originally purchased we did not expect to have scalability issues. One of the challenges we ran into was that the server had two main roles – collecting data and providing data. The later refers to the fact that we share all data with users via a website and a display in the building lobby.

A few months into the deployment, the amount of gathered data became large enough that displaying a week-long timeline for a single device would take prohibitively long; generating a summary graph for all devices on the fly was out of the question. Thus, PowerNet periodically runs a set of data summary calculations. For example, every 5 minutes the server establishes what meters are reporting, takes the fine-grained data, averages it, adds it up, and produces a graph like Figure 3.

A couple of times we observed that the server load was so high due to nightly scheduled backups and both MySQL and rsync experiences issues. The scalability and performance issues we have observed so far have prompted us to consider a number of back-end improvements. These include partial database backups via the binary log option in MySQL and incremental pre-calculations to summarize data. In the future, we plan to extend the system by one or more additional servers and distribute the load and backup responsibilities.

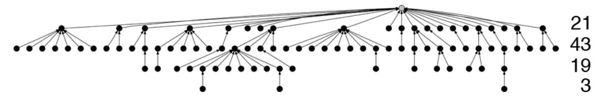

Figure 13. Logical topology of the wireless network. The root of the tree is on top, and the number of nodes at each level is shown.

6 Wireless Meter Network

The prior section examined our experiences with the overall PowerNet deployment. This section dives into the performance of the wireless network, specifically the Collection Tree Protocol. We chose CTP because it is the standard protocol in TinyOS 2.x and extensive testbed experiments over the scope of hours indicate that it is robust and efficient [7]. This section examines whether CTP exhibits similar performance and behavior in an operational sensor network over a three month period, a timescale two orders of magnitude larger than the prior study. Figure 2 shows the physical map of the wireless deployment, while Figure 13 presents a snapshot of the logical topology as constructed by CTP.

Because the wireless network does not have a wired back channel, we add instrumentation to CTP to report statistics such as data transmissions, retransmissions, and receptions, beacon transmissions, and parent changes every 5 minutes. PowerNet uses 802.15.4 channel 19, which overlaps with heavily used WiFi channel 6. We chose this so we would not interfere with research using quieter channels (e.g., 25 and 26) and so that we could measure CTP in a less forgiving environment.

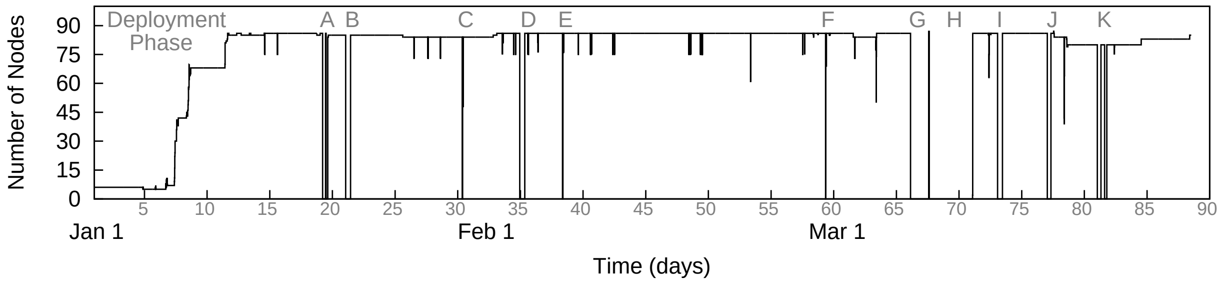

Figure 12. Number of nodes from which packets were received at the basestation during the deployment.

Label	Date	Duration	Description
A	Jan 19	9 hrs	Building power outage
			MySQL recovery
B	Jan 21	10 hrs	Backend maintenance/backup
C	Jan 30	1 hr	Basestation maintenance
D	Feb 4	9 hrs	Basestation software failure
E	Feb 8	1 hr	Backend maintenance
F	Feb 28	0.5 hr	Backend maintenance
G	Mar 8	34 hrs	Backend disk failure
H	Mar 9	83 hrs	Backend disk replacement
I	Mar 14	9 hrs	Basestation buffering
J	Mar 18	7 hrs	Basestation buffering
K	Mar 22	4 hrs	Backend RAID1 rebuild

Table 8. System Outages

Nodes	85
Path Length	1.84
Cost	1.91
Cost/PL	1.04
Churn/node-hr	5.04
Avg. Delivery	0.969
5th % Delivery	0.789
Loss	Retransmit

Table 9. High-level CTP results, following the metrics in the CTP paper [7]

6.1 Summary of Results

Overall, the backend collected 85.9% of the expected data. Of the 14.1% of missing data, 8.2% is due to backend failures, such as whole-building power outages or server disk failures. This type of failures also affected data from the wireless meters and utilization sensors. Of the remaining 5.9%, we approximate that 2.8% is due to users taking meters offline by unplugging them: the remaining 3.1% of data losses are due to CTP.[2]

Sifting through CTP's periodic reports, we find weekly and daily cycles of topology adaptation that correspond to human activity in the building. These periods of adaptation see a significant increase in control traffic as well as increased path costs. In the middle of the night, the average cost (transmissions/delivery) of the network is just under 2, while during the day it can climb as high as 6. We find that CTP's datapath validation leads to a tiny fraction (1 in 20,000) of packets taking 10-100 times as many hops as normal, as they bounce through the topology repairing loops. Finally, we present a bug we discovered in CTP's link estimator where nodes are unwilling to route through a rebooted node for a very long time, which can be disastrous if a base station reboots. We present a fix to the bug, which the CTP maintainers have incorporated into the recent TinyOS 2.1.1 release.

[2] We assume the CTP delivery for the days 39-59 to be representative for the full deployment period.

6.2 System Uptime

Figure 12 shows a 90-day trace of the number of connected wireless meters reported for each 15-minute period. Over the 90 days, the network experienced 11 network-wide outages in data logging, labeled (A–K). Table 8 describes each outage, including whole-building power loss, backend downtime maintenance, disk failures, and gateway PC software failure. Overall, the backend was down for days, giving PowerNet an uptime of 91%.

Small dips in the number of reporting nodes (e.g., the two dips at 15 days) represent logging delay due to MySQL buffering. These delays do not denote data loss.

While the high point of the plot remains stable (e.g., between points D and F), it does vary. For example, a week around K (days 77-84) shows 8 nodes stopped reporting. This is not a network failure: the eight nodes were all in the same room (the labeled room in Figure 2). The 8-node outage occurred when the room was repainted and all computing equipment was unplugged and moved. Other, smaller dips represent users unplugging meters. Generally speaking, no data delivery outage observed was due to a failure in CTP or the wireless meter network. This deployment data validates prior testbed results on CTP's robustness [7].

6.3 CTP Performance

To isolate CTP's performance from network and node downtime, all of these following results are from a 20-day period in February (days 39-59 in Figure 12.) CTP's behavior in this particular 20-day period is representative of the rest of PowerNet's lifetime after deployment.

Table 9 shows high-level results following the methodology used in the CTP publication [7]. The PowerNet network behaves differently than any of the studied testbeds. On one

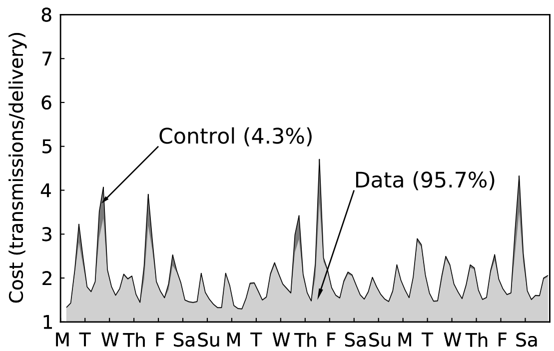

Figure 14. Average packet delivery cost over 20 days. Weeknights and weekends show lower cost due to the availability of more efficient and stable paths. Cost of 1 is optimal.

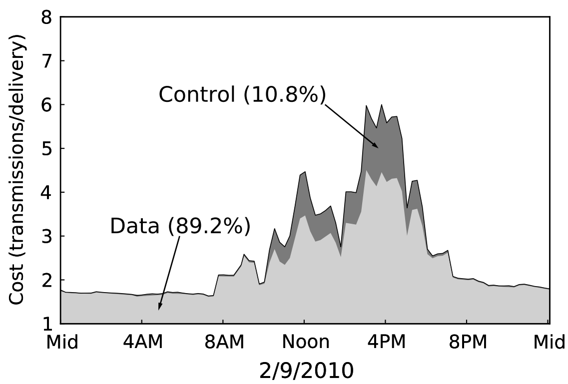

Figure 15. CTP's packet delivery cost over one day; a value of 1 is optimal.

hand, its cost per path length of 1.04 indicates that intermediate link are rarely used. (on average out of 104 packets only 4 were retransmission), making it similar to testbeds such as Mirage. On the other hand, its high average churn rate of 5.04 per hour makes it similar to harsher testbeds such as Motelab. This indicates that while PowerNet has many high quality links, those links come and go with reasonable frequency.

CTP's average delivery ratio was 96.9% and only five out of the 85 nodes reported delivery ratio below 90%. Two of these nodes were near many other wireless nodes, while another two were in the corner, possibly using longer links. The principal cause of packet loss is retransmission failure: CTP drops a packet after 30 attempts to transmit it on a single link.

Figure 14 shows CTP's average cost (transmissions/delivery) over a 20 day period divided into data transmissions and control beacons. While the average cost is below 2, the middle of workdays can see the cost climb as high as 4.5, as the network adapts to topology changes. Figure 15 shows the same plot for a single work day. On this day, the cost rises as high as 6, and control beacons constitute 10.8% of the packets sent. The peak in Figure 15 is higher than those in Figure 14 due to longer averaging intervals.

CTP's control traffic rate is bimodal. While 85% of nodes

Hops	1	2	3	4	5	6-20	20-190
Fraction	39%	42%	16%	2.6%	0.57%	0.039%	0.0051%

Table 10. Distribution of CTP packet path lengths.

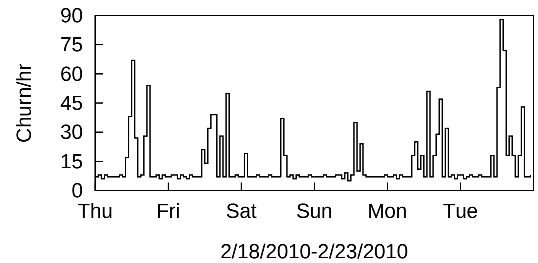

Figure 16. Churn for one node over a six day period. Weekday afternoons and evenings show higher churn than weeknights and weekends.

send a beacon every 15 minutes or less, 15% of the nodes send over ten times this many. As these high-traffic nodes are typically also forwarding many data packets, CTP's uneven control load can impose an even higher energy burden in low-power networks and harm network lifetime.

6.4 Daily and Weekly Cycles of Churn

Table 9 shows that CTP observes significant parent churn even during a stable, long-term deployment. This churn could be because CTP topologies are inherently unstable even in a stable environment, or because the underlying environment itself is unstable. Figure 16 shows a 14-day time series of one node's parent change rate with clear daily and weekly trend. During working hours, the node experiences much higher churn, up to 90 parent changes/minute. Furthermore, the peaks on the weekend are shorter and smaller than weekdays. In the absence of human activity, churn is fairly constant, at approximately 6 parent changes per hour.

6.5 Datapath Validation

CTP data packets contain a Time Has Lived (THL) field, which increments on each hop. Measuring THL at the gateway allows us to measure how many hops packet traverse in the network. Table 10 shows a distribution of path lengths. Most packets fall in the range of 1-5 hops, one in 2,600 packets takes 6-20 hops, one in 20,000 packets takes 20-190 hops, and with one packet out of over 15,400,000 taking 190 hops.

The small percent of high THL packets stem from CTP's datapath validation algorithm. CTP uses data packets to validate and repair its topology. When a node detects the cost gradient is not decreasing, it sends beacons to repair the topology but forwards packets normally. This algorithm allows CTP to quickly detect potential loops in the network, but does not necessarily repair them quickly. Correspondingly, some packets take a very large number of hops to repair. The longest loop was 190 hops and was repaired in 7.7 seconds.

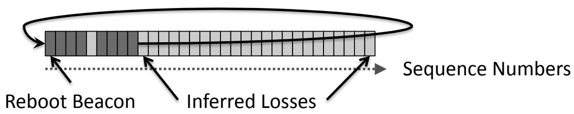

Figure 17. Visual depiction of CTP link estimation bug. On reboot, the link estimator infers a sequence number 0 packet as a long string of failures, raising the link cost high enough that CTP will not use it.

6.6 Duplicate Suppression

We find that overall 1.7% of the packets received at the basestation were duplicates. Packets from eight nodes had a duplication rate above 3.7%. During our 90-day deployment, due to misconfiguration, we deployed two nodes with ID 185 in two different areas of the network. The two nodes continue to report readings to the basestation but there are twice as many packets logged at the server. These packets elude CTP duplicate suppression due to two reasons. First, these two nodes often do not share a path. Second, the packet signature used for duplicate detection includes node ID, sequence number, and number of hops but the latter two are rarely the same between packets of the two nodes.

6.7 Link Estimation Bug

We encountered one bug while deploying CTP that existed in CTP's four bit link estimator (4B) [12]. We observed the bug during test deployments in December of 2009 and it did not affect the real 90 day deployment presented here.

The bug occurs when a mote reboots and other motes do not choose the rebooted mote as a next hop for many hours. In the case when the CTP root reboots, this causes the entire topology to collapse and encounter the count-to-infinity problem.

The bug stems from how the link estimator handles beacon packets. When CTP sends a beacon, the link estimator adds a header and a variable number of footer entries. The header contains a sequence number, so nodes can infer losses by sequence number gaps. The arithmetic, however, is such that if a node reboots and sends sequence number zero, nodes assume that all packets between the last one heard and 0 were lost, as shown in Figure 17. Such a long string of losses causes the link cost to climb far above the cutoff threshold CTP will use. The only thing that can bring the link cost down is a long series of received beacons. However, CTP's adaptive beaconing means that it can take hours to days for a long enough sequence.

This bug is not particular to the root. Nodes that reboot will not be chosen as parents. If a network is dense enough, the removal of one parent does not greatly harm the topology, as nodes can route around it. It is worth noting that the CTP publication evaluated the effect of node failures on performance, but not reboots.

We fixed this bug by capping the number of losses a sequence number gap can infer to 10. Doing so caps how far in history CTP considers sequence numbers, causing it to lend more weight to the recent reception than the prior losses. Incorporating this fix allows CTP to operate properly in the face of even somewhat common node reboots. The CTP authors have incorporated our fix into the standard implementation.

7 Discussion

PowerNet's extensive power and utilization measurements reveal how different parts of a computing infrastructure contribute to total power cost. This section discusses several approaches which can help reduce power consumption.

7.1 Interventions

While energy-efficiency improvements have the greatest potential to reduce power consumption, educating users should not be under-estimated. Section 3 showed that small changes in how we use LCD screens can lead to 20% savings. We have found that informing users about the power draw of their monitors and giving suggestions on how they can conserve energy has affected behavior positively.

In the future we anticipate expanding these efforts in several ways. One is to have an interactive display that allows building occupants to dig through the data, exploring it in a way that interests them. Such engagement with real-world data brings attention to energy consumption. In addition, we plan to make individual data available to users who volunteer to participate in the PowerNet monitoring. Power data will always be tied back to utilization to remind people of situations in which energy is wasted.

7.2 Policy Changes

In addition to educating individual occupants, our work has provided insights to the administrative and IT staff in the building. Simply providing detailed data of power usage has prompted the staff to think about possibilities for savings.

For example, Section 3 briefly mentioned that staff machines are required to be powered on at night so data backups can complete. These backups can also be observed in Figure 10 by noticing the daily traffic spikes, for example the ones shortly after midnight. We learned that different groups of machines had different start backup times but no machine had to be on for more than one hour. We pointed out that powering staff machines 24-7 was wasteful since they were never needed for more than approximately 12 hours a day. The suggestion we heard back was that backups could occur during the lunch hour. Instead, we plan to propose that Wake-on-LAN is used in conjunction with the backup system. The scripts that currently run can be modified to wake a machine before the backup and put it back to sleep one hour later. The current backup policy is causes at least 30 machines to waste 32 kWh every day, costing $130 a month.

7.3 Technical Alternatives

By far, the most effective way to reduce energy consumption of computing systems is to shut them down when they are not needed. Section 3 showed that personal computers constitute 16% of the building's energy consumption, while Section 4 showed they are rarely used for more than 12 hours a day. The energy consumption data suggests that turning idle

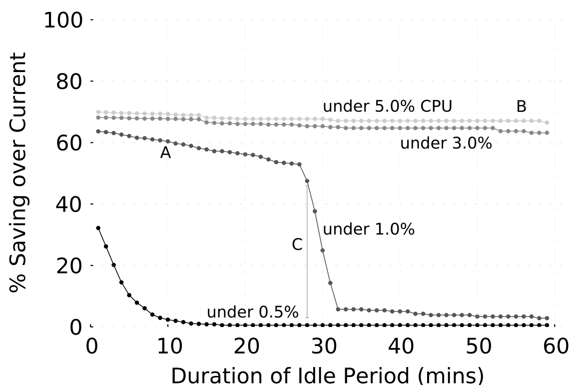

Figure 18. Different interpretations of 'idle' lead to varying energy savings. Even conservative estimates such as 25 minutes at CPU of 1% or lower can lead to over 50% reduction on energy use.

Energy Consumption	High-end machine	Low-end machine
current	38.6 kWh	16.8 kWh
idle sleep	14.6 kWh	6.7 kWh
power proportional	2.0 kWh	1.2 kWh

Table 11. The same user workload can result in different energy consumption based on the type of machine and sleep policies being used. A less over provisioned, low-end dekstop with idle sleep mode can reduce consumption by 6 times over a more powerful machine that never sleeps.

machines off could reduce the building's energy consumption by 8% – $3,600 a month.

To illustrate this point we analyze a week-long trace of power and CPU for a student desktop. This is the same data trace presented earlier in Figure 9. How would the desktop's energy consumption change if the machine was in sleep state, drawing 5 watts, when it was idle? We define idle as 'using W% CPU or less for X minutes and longer'. Figure 18 shows the calculated saving for different values of X and W.

For example, if the machine is put to sleep after 10 minutes of CPU utilization under 1% (Point A in the Figure), the energy consumption over a week will be 60% less if the machine stayed awake. Defining 'idle' as CPU of 5% or lower results in 70% saving for even the longest timeout values (point B.) For this specific machine, the difference between staying under 0.5% and 1% CPU is significant – with a threshold of 1% CPU usage we can still save almost 50% of energy consumption for idle periods of 30 minutes of fewer.

Taking the analysis one step further, we compare a smart idle approach to an energy proportional system. The results in Table 11 are from the same data trace as the previous experiment. This desktop consumed 38.6 kWh in one week; if it were put to sleep after 5 minutes of CPU of 1% or less, it would have consumed only 14.6 kWh. With an energy proportional system consumption is only 2kW.

Energy proportionality in desktops is hard to achieve because components such as processors, disks, and graphics cards have high baseline power draw. So is there another way to get close to the low consumption that proportionality achieves? Since desktops are highly over-provisioned in many cases a less-powerful and less power-hungry machine could match user workload better. Table 11 present calculation that apply the first desktop's utilization trace on a machine that draws only about half of the power but has the same processing resources.

Equipment	Purpose	Power
26 SunRay Clients 1, 1G, 2FS	client	26 x 30 W
2 SunFire X4200 Server	server	2 x 550 W
1 SunFile V420	file system	1 x 320 W
2 Sun StorEdge T3	storage	2 x 450 W
	Total:	**3100 W**

Table 12. Terminal-style clients with Sun servers are an alternative approach to desktop-centered computing. Preliminary analysis shows that such a setup is as or more efficient than individual desktops.

The conclusion is that in many cases there are simpler changes that can lead to great savings. For example, by choosing a lower-power machine and putting it to sleep, one can reduce consumption by 6 times without the complexity of designing an energy proportional machine. As Section 4 showed, desktops are rarely fully utilized so CPU needs will be met even by a lower-power machine.

The low utilization of desktops leads to yet another computing alternative. Currently, most users work on dedicated machines, physically present in the office. In addition to those, many people also connect to server machines via SSH.

A different academic department at a European university does things in a less distributed way. They have set up a Sun client-server system where clients have minimal processing and storage and instead act as dumb terminals. Table 12 shows the equipment used to support the computing of 26 people – 1 professor, 3 post-docs, 4 admins, and 18 PhD students. The power draw for the Sun server setup is approximately 120 watts per person and is believe to also be under utilized. among other benefits a thin-client approach will also take advantage of high network bandwidth availability.

8 Related Work

Historically, the phrase "energy metering" has referred to the coarse-grained measurements provided by spinning analog dials outside a structure, read once a month. New meters monitor consumption at the individual device, power strip, or whole-house levels [1, 5, 10], and some log data through wired Ethernet, low-power wireless, or even the power lines they measure. PowerNet uses a modified version of the ACme meter design [14], which provides wireless connectivity and open access to both software and hardware.

Power monitoring deployments differ by the scale at which data is measured. The Green Soda project [14] takes a fine-grained approach. It monitors about 30 individual devices and several power strips, along with light sensors, which the infer power consumption of overhead lighting. The Green Soda project demonstrated the feasibility of an indoor wireless monitoring infrastructure. The similar PowerNet project [15] presents initial insight into the power and utilization of computing systems, with mostly wired meters.

PowerNet builds upon the Green Soda and PowerNet projects in several ways. The system measures more devices and a greater variety of computing devices, over a much longer time period. The addition of utilization meters enables correlated power and utilization measurement, which enables us to draw conclusions about efficiency, not just the breakdown of energy usage. The wireless deployment is unusally dense, and our experiences with its performance and operation can provide guidance for future power monitoring efforts, as well as indoor sensor deployments.

Other green computing projects have looked into the challenges of visualizing power data and presenting it to building residents. Energy dashboards [3, 4] and websites [2, 9] summarize and compare power usage data in order to encourage savings. Many universities have taken advantage of dashboard software to educate students living in dorms, generally with measurements at the granularity of one floor or the whole building. The Energy Dashboard Project at UCSD [8] covers academic buildings with one to four aggregate meters in each building. Such data is useful when comparing power consumption between buildings and looking for high-level trends in the data. However, aggregate power goes not identify the parts of the building or the types of devices that are wasting energy. While not the focus of this paper, PowerNet has also joined in the effort, with a website and display in the lobby. Unlike other dashboards, ours includes utilization data to highlight wasted energy.

9 Conclusion and Future Work

One over-arching question drove the PowerNet deployment: how can fine-grained power and utilization data create a high-level, building-scale, actionable understanding of the usage and efficiency of the our computing infrastructure?

The first challenge was that of collecting data – how many and what type of sensors are needed, and where they should go? Throughout the deployment we learned that wired sensors have their place for sparse measurements, but that a dense network of open-source wireless sensors avoids the unexpected practical issues of installation and debugging. We found that CTP performance degrades during business hours, but provides reasonably high transmission rates and robustness for a large deployment in a previously-untested indoor setting. The current system has collected over 150 gigabytes of data, but is still far from perfect. Future priorities are include adding 300 wireless meters to cover more servers and other equipment, and installing more utilization sensors.

Once fine-grained data is in, the next challenge is to derive an accurate breakdown suitable for making high-level observations about classes of devices. The aggregate measurements of 138 power meters amount to only 2.5% of the building total. However, by learning device totals and distributions from other sources, including surveys, observations, and IT database records, then cross-correlating these with power data, one can construct a reasonably accurate quantitative breakdown. Precise extrapolation requires knowledge of how individual devices within a class compare. For example, desktop power shows high variation, up to 10x, and thus dense instrumentation is needed. On the other hand, network power draw is constant over time so only a few sensor readings suffice.

We find that desktops and servers account for 46% of the building's electricity consumption, monitors account for 7%, and networks for 3%. While these numbers might differ for other office buildings, our methodology and high-level insights will remain valuable. They guide our understanding of how to have meaningful impact on reducing energy consumption.

Therefore, the final challenge is turning quantitative analysis into qualitative comparisons, recommendations for computer system design – and even purchasing guidelines. On this front, we claim no complete answers, only initial insights. The energy breakdown shows where to focus efforts, while the correlated power and utilization measurements highlight areas of inefficiency. Specifically, we show that determining idle state and transitioning PCs to a low-power mode can have a dramatic impact. Another example is monitors; the data showed a harmless way to save energy. The fact that a few offices have actually our suggstions, resulting in energy savings encourages us to continue building out the deployment and mining the data.

10 References

[1] Arch Rock. www.archrock.com.

[2] Google Power Meter. www.google.org/powermeter.

[3] Lucid Design Group Building Dashboard. www.luciddesigngroup.com.

[4] Onset Data Loggers. www.onsetcomp.com.

[5] Plugwise. www.plugwise.com.

[6] RPL: IPv6 Routing Protocol for Low power and Lossy Networks. IETF Draft, http://tools.ietf.org/html/draft-ietf-roll-rpl-07.

[7] TEP 123: Collection Tree Protocol. http://www.tinyos.net/tinyos-2.x/doc/.

[8] UC San Diego Energy Dashboard Project. energy.ucsd.edu.

[9] Visible Energy Inc. www.visiblenergy.com.

[10] Watt's up internet enabled power meters. `https://www.wattsupmeters.com/secure/products.php`, 2009.

[11] P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler. A building block approach to sensornet systems. In *In Proceedings of the Sixth ACM Conference on Embedded Networked Sensor Systems (SenSys'08)*, 2008.

[12] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis. Four-bit wireless link estimation. In *Hotnets-VI*, Atlanta, GA, 2007.

[13] J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the Second International Conferences on Embedded Network Sensor Systems (SenSys)*, 2004.

[14] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and implementation of a high-fidelity ac metering network. In *Proceedings of The 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '09)*, San Francisco, CA, USA.

[15] M. Kazandjieva, B. Heller, P. Levis, and C. Kozyrakis. Energy dumpster diving. In *Workshop on Power Aware Computing and Systems (HotPower'09)*, 2009.