Energy Aware Network Operations

Priya Mahadevan, Puneet Sharma, Sujata Banerjee, Parthasarathy Ranganathan
HP Labs
Email: $\{$*firstname.lastname*$\}$*@hp.com*

***Abstract*—Networking devices today consume a non-trivial amount of energy and it has been shown that this energy consumption is largely independent of the load through the devices. With a strong need to curtail the rising operational costs of IT infrastructure, there is a tremendous opportunity for introducing energy awareness in the design and operation of enterprise and data center networks. We focus on these networks as they are under the control of a single administrative domain in which network-wide control can be consistently applied. In this paper, we describe and analyze three approaches to saving energy in single administrative domain networks, without significantly impacting the networks' ability to provide the expected levels of performance and availability. We also explore the trade-offs between conserving energy and meeting performance and availability requirements. We conduct an extensive case study of our algorithms by simulating a real Web 2.0 workload in a real data center network topology using power characterizations that we obtain from real network hardware. Our results indicate that for our workload and data center scenario, 16% power savings (with no performance penalty and small decrease in availability) can be obtained merely by appropriately adjusting the active network elements (links). Significant additional savings (up to 75%) can be obtained by incorporating network traffic management and server workload consolidation.**

I. INTRODUCTION

Energy efficiency has become crucial for all industries, including the information technology (IT) industry, as there is a strong motivation to lower capital and recurring costs. With the advent of the Cloud Computing model, large data centers are being built that consolidate processing and storage for a large number of services accessed over the Internet or enterprise networks. In such environments, the initial focus on energy efficiency has been on cooling and server power[1] management [1]–[3] and significant advances have been made in these areas. Recent studies have shown that networking devices account for about 15% of a data center's total energy consumption [4]. So far, scant attention has been paid to make networking in enterprise and data center networks more energy efficient. In this paper, we focus on energy savings algorithms for networking components in enterprise and data center networks, that typically are under the control of a single administrative authority, and thus making it possible to apply network-wide energy saving schemes.

Ideally for power-efficiency, devices should consume energy proportional to their load [5]. The majority of the network devices deployed today are far from being energy proportional and provide a very limited set of knobs to control their power consumption [6]. In this paper, we focus on how to achieve energy efficiency from these non-energy proportional devices. We propose several energy saving algorithms for efficient configuration and management of data center networks. We simulate the effects of these algorithms on a real Web 2.0 workload from an operational data center.

We find that 16% network energy savings can be obtained with no performance penalty and slight decrease in system availability. Significant additional savings up to 75% can be achieved by using traffic management and server workload consolidation in the data center. We also quantify the performance penalties of deploying these algorithms. While our overarching research goal is to ultimately influence the next generation of router/switch hardware to make them more energy-aware, we would also like to introduce energy awareness in the operation of a large legacy base of equipment currently deployed. Thus we attempt to implement our algorithms with existing control knobs that are readily available in networking devices in operation today.

The rest of this paper is organized as follows: We provide an overview of data center architecture and energy consumption of networking components in Section II. We describe our algorithms in Section III. Our workload and results from our schemes are detailed in Section IV. We describe how we can implement our algorithms in Section V. We discuss related work in Section VI and finally conclude in Section VII.

II. POWER PROFILE OF NETWORK DEVICES

As an initial step to understanding the energy consumption patterns of a variety of networking devices, we conducted a detailed power instrumentation study within our own data center [6] in order to identify various control knobs that can be tuned to save networking power. The first knob is to disable a switch port when it is not forwarding any traffic. Next, we can dynamically set the forwarding capacity of individual ports based on its load. Since power consumed by a port is dependent on its speed, the power savings in this case depend on the load and the port's operating speed. In case of devices with multiple linecards, another control knob is to turn off the linecards that have no active ports. We note that not all switches support the option of turning off a linecard. Finally, we can completely power off a switch that is not being used.

In our previous study [6], we analyzed power measurements obtained from a variety of switches, as a function of switch configurations and traffic flowing through the switch. Based on our analysis, we developed a power model to estimate the power consumed by any switch, $Power_{switch}$. Our linear power model is defined as $Power_{switch} = Power_{chassis} + num_{linecards} * Power_{linecard} + \sum_{i=0}^{configs} numports_{configs_i} * Power_{configs_i}$

---

[1] We use power and energy management interchangeably in this paper. There is a distinction between power management for heat density versus electricity costs; however in this paper, we do not distinguish between these two issues.

Configuration	Rack switch (in Watts)	Tier-2 switch (in Watts)
$Power_{chassis}$	146	54
$Power_{linecard}$ [a]	0 (included in chassis power)	39
$Power_{10Mbps}$ (per port)	0.12	0.42
$Power_{100Mbps}$ (per port)	0.18	0.48
$Power_{1Gbps}$ (per port)	0.87	0.9

[a] Both of these switches do not support the capability to power off individual linecards, hence we add the power cost of linecards to the switch's total power consumption in our experiments.

TABLE I

POWER CONSUMPTION SUMMARY FOR ENTERPRISE SWITCHES.

$Power_{chassis}$ is the power consumed by the switch's chassis; $Power_{linecard}$ is the power of a linecard with no active ports, and $num_{linecards}$ is the actual number of cards plugged into the switch. Variable *configs* in the summation represents the possible configurations for a port's linespeed (typically it can be 10 Mbps, 100 Mbps or 1 Gbps), $Power_{configs_i}$ is the power consumed by a port running at linespeed i; $numports_{configs_i}$ is the number of ports at linespeed i, where i can be 10 Mbps, 100 Mbps or 1 Gbps.

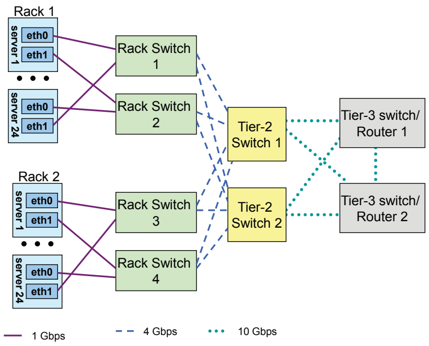

Fig. 1. Data center architecture

With a basic understanding of power consumption in a switch, we next describe how switches are typically connected in a data center network. Given the mission critical nature of jobs, enterprise networks and in particular data center networks are typically required to be always-on. Hence network topologies are designed to be over-provisioned and highly redundant. Though these topologies provide fertile grounds for saving energy, any such scheme has to be carefully implemented so as not to compromise the availability and performance of the system. As an example, we show the architecture of the data center in our lab in Figure 1. We call this topology a 1-redundant tree due to the extra (*i.e.* 1) redundancy that exists at each level of the tree. As shown in the picture, each server has two network interface cards that connect to different switches in the rack; we call them rack switches. However, at any given instant, only one rack switch is responsible for forwarding all traffic to and from each server, thus ensuring load balancing amongst the rack switches. At the next level, each rack switch is connected via a trunk of four 1 Gbps links each to two tier-2 switches. These tier-2 switches connect to tier-3 switches, that perform both L2 forwarding and L3 routing. Our topology is typical of most data centers, though the exact link bandwidths might vary. In Table I, we list the power consumed by each configuration taken from actual switch measurements in our data center. The storage traffic goes over a separate storage area network (SAN) and we do not consider SAN in this paper.

III. ENERGY SAVING SCHEMES

We now present three schemes for reducing the energy consumed by networking equipment in any network with a single administrative control domain. Besides adapting the network to the load (using the knobs described in the previous section), networks can be made more energy-efficient by energy-aware job-allocation algorithms. In such schemes, the placement algorithms take network traffic specifications of the job, the current network utilization, and the topology into consideration before assigning various servers for a job.

In the context of a data center network, we envision a centralized network power controller program running on a server within the data center. For very large networks, a distributed power controller may be required; here we focus on a centralized controller. This program gathers traffic data from all the switches in a data center using SNMP or other tools, thus computing the utilization of each port on all the switches. Based on this information, the controller communicates with all the switches and performs actions such as turning off unused switches, disabling unused ports and adapting link capacity. We discuss deployment considerations for our algorithms in Section V.

Link State Adaptation (Strawman or LSA): Similar to ideas described in [7]–[9], in this scheme, the power controller uses information about traffic on each link and adapts its state accordingly. Typically, each link can operate in four states, namely, *disabled*, 10 Mbps, 100 Mbps and 1 Gbps. This basic scheme merely ensures that the traffic can be accommodated without regard for performance and availability.

Network Traffic Consolidation (NTC): We can adopt a traffic engineering approach to route traffic such that it is consolidated on fewer links (and switches), while some of the non-utilized links (and switches) can be disabled. This approach reduces energy consumption significantly by removing all redundancy in the network. The energy consumed is the minimum required to support the offered network load, but it comes at a great cost to reliability as there are no redundant paths in the topology. In our topology (Figure 1), the 1-redundancy at every level can be reduced to 0; exactly 1 switch can be operational for a rack, with all the servers on a rack transmitting their traffic to the one operational switch, while the other switch can be turned off. While not practical, we use this case to show the trade-offs between power savings and availability. Clearly, there will be occasions when one can be traded for the other. In addition to this 0-redundancy (but most power efficient) approach, we explore the space in between where some energy efficiency can be sacrificed for more redundancy in the overall topology; we quantify the tradeoffs for different levels of energy efficiency and system redundancy.

Server Load Consolidation (SLC): Current job-allocation algorithms do not take network traffic and topology into consideration and fail to optimize for network energy consumption. An indirect way to consolidate network traffic onto fewer links and allow the controller to turn off non-utilized ports (and switches) is to migrate jobs, so a fewer number of servers are being used. We need to ensure that server resources such as CPU and memory are adequate to handle the assigned jobs. After performing server load consolidation, we can further reduce energy consumption by resorting to network traffic consolidation as well (NTC scheme).

In all our schemes above, the energy savings come at the cost of availability and reliability. For example, if a link of capacity 1 Gbps has 5 Mbps traffic flowing through it, rate adapting this link from 1 Gbps to 10 Mbps will save energy, but might lead to increase in latency due to queuing delay. To mitigate this decrease in performance, we can add a constraint to ensure that a link's utilization never exceeds a certain threshold before adapting its rate. Thus, in all three of the above schemes, we can incorporate Service Level (SL) awareness by adding constraints to ensure that a minimum performance is achieved. We term these SL-aware LSA, SL-aware NTC and SL-aware SLC schemes respectively.

For instance, a SL-aware LSA scheme that incorporates performance guarantees will ensure that each link's delay is kept under a threshold by putting an upper bound on the utilization (say, 70%) and also make sure that there is redundancy in the system to counter failures. To address fault-tolerance, we can ensure that at least one other redundant path exists (even if it is at the lowest bandwidth) before turning off links/switches. We can similarly apply this service level awareness to NTC and SLC schemes as well.

In the next section, we describe the details of a Web 2.0 trace that we collect from a real data center and simulate our schemes on this workload. We compare the schemes with respect to energy saved, system availability and performance.

IV. CASE STUDY: RESULTS FROM A REAL WEB 2.0 WORKLOAD

In order to evaluate the energy savings and the performance impact of our schemes, we collect system and network traces from a production data center hosting an e-commerce application. The servers in the data center are organized in a tiered model as application servers, file servers and database servers.

We use the System Activity Reporter (sar) toolkit available on Linux to monitor CPU, memory and network statistics including the number of bytes transmitted and received from 292 servers. Our traces contain statistics averaged over a 10-minute interval and span 5 days in April 2008. Each server has two 1 Gbps network cards. The servers typically have quad-core processors and RAM varying from 4 to 16 GB. Of the 292 servers, 193 servers have 4 GB RAM, 69 servers have 8GB RAM and 30 servers have 16GB RAM. Our workload is memory-intensive, with 130 servers using 90% or greater of their memory. On the other hand, 64 servers use less than 40% of their memory. Both network traffic and CPU usage is light, with all four processors on all servers being at most 10% utilized at any given time. The aggregate traffic through all the servers varies between 2 and 12 Gbps at any given time instant as shown in Figure 4.

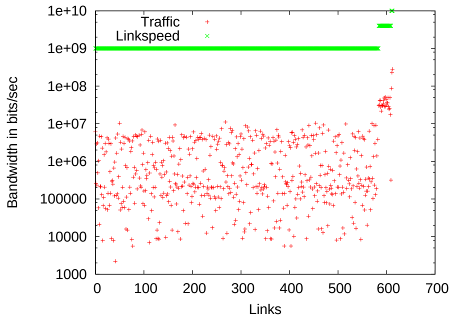

Fig. 2. Observed traffic versus linkspeed for all links in the original topology. Observed traffic is averaged over the entire trace duration.

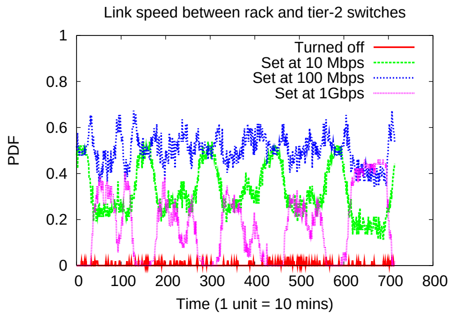

Fig. 3. Fraction of links between rack and tier-2 switches set at different linkspeeds as a function of time for our baseline LSA scheme

This production data center topology is similar to the one we describe and power instrument in Figure 1. To understand the effects of LSA, NTC and SLC schemes with respect to energy savings and availability, we simulate each scheme on the above workload. The 292 servers (connected as shown in Figure 1) are assigned to 13 racks — each rack containing up to 24 servers and exactly 2 rack switches. The first twelve servers in a rack have their primary network interface connected to the first rack switch and their secondary interface connected to the second rack switch. This connection is reversed for the remaining twelve servers. At the next level, each rack switch connects to two tier-2 switches, via a 4 Gbps (four 1 Gbps trunked) link to each tier-2 switch. We have two tier-2 switches in our topology; both tier-2 switches connect to all 26 rack switches. Further up the hierarchy, we have two tier-3 switches that act as root switches/routers for transmitting packets in and out of the data center. Each tier-2 switch has a 10 Gbps (ten 1 Gbps trunked) connection to each tier-3 switch. Both the tier-3 switches also connect to each other via a 10 Gbps (ten 1 Gbps trunked) link.

Each rack switch has 48 ports of 1 Gbps capacity each. Each tier-2 switch has a 6-slot chassis, each of which can be fitted with linecards having 24 ports of 1 Gbps linespeed each. In our simulations, we use all 6 linecards for a total of 144 ports. For the connection between a tier-2 switch and a tier-3 switch, we use ten separate 1 Gbps ports to form a single 10 Gbps link (as all the ports on tier-2 switch have only 1 Gbps capacities). Based on this architecture, each rack switch has 32 active ports, 24 of which are connected to servers and the remaining to the two tier-2 switches. Of the 144 ports in each tier-2 switch, 124 are active; four ports are linked to each rack switch, and ten ports are used for connecting to each tier-3 switch.

We use the linear model presented in our previous work [6] to estimate the power consumed by each switch depending on the number of active ports and the linespeed of each port. The actual power consumed (in W) by individual switch components is summarized in Table I. We note that our tier-2 switch model in the data center does not support the capability to power off individual linecards. Hence, for a tier-2 switch that is powered on, we include the cost of all six linecards (234 W) while computing the switch's power consumption. In all our experiments, the energy savings come from rate adapting a port, disabling a port, as well as completely turning off a switch.

From our traces, we find that approximately 70% of the traffic is internal to the data center, while the remaining 30% is external to the data center. Thus, for simulating communication patterns for our energy saving schemes, for each server, we choose 70% of the outgoing traffic on each interface to be randomly distributed to other servers within the data center and the remaining 30% of the traffic is sent to one of the tier-3 switches to be sent outside the data center. In Figure 2, we plot both the actual capacity as well as the average traffic through each link in our data center, to quantify the high degree of over-provisioning that currently exists.

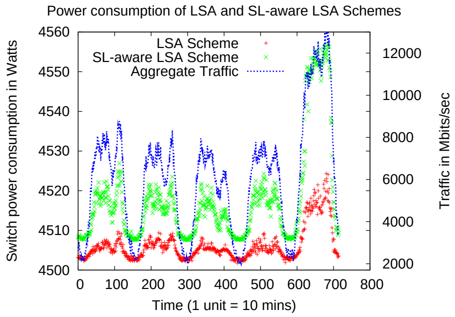

Fig. 4. Power consumption for baseline LSA scheme vs SL-aware LSA scheme. In the SL-aware LSA scheme, utilization threshold of any link is 70%, while an alternate path always exists for any pair of servers. The aggregate traffic through all 292 servers over the 5 day period is plotted against the secondary y-axis.

Based on this communication pattern, we compute the total number of active ports and the corresponding port utilization on each rack and tier-2 switch in the data center. Next, we compute the total power consumed by all the rack and tier-2 switches over the trace duration. We ignore the power consumption of the tier-3 switches in our calculation as we do not apply our schemes to these switches. In the baseline case, when no power saving scheme is applied, the power consumption just for the switched network is 5402 Watts and it remains constant over the entire trace duration.

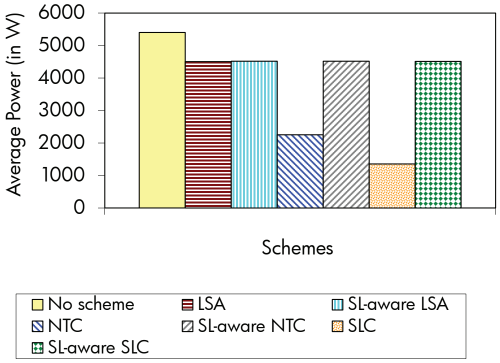

Fig. 5. Comparing the average power consumed for all our schemes. In the SL-aware version of the three schemes, each link's utilization is below 70% and an alternate path between every host pair always exists.

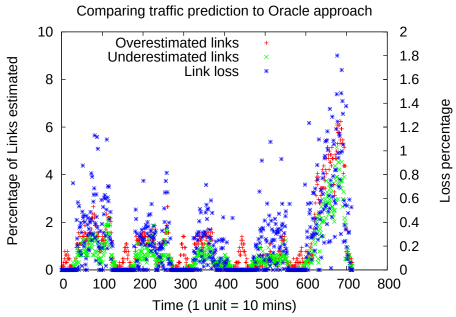

Fig. 6. Link speed settings for NTC scheme using predicted traffic

We note that in order to implement the various schemes in realtime, a system controller will require timely and perfect information about the traffic load. In this section, we assume the existence of an *Oracle* which has perfect information about the upcoming traffic and is queried by the controller before adjusting the switch and port configurations. Our goal is to estimate the energy savings for the different schemes and understand their availability and performance tradeoffs. We discuss traffic prediction strategies to replace the *Oracle* in Section V.

We summarize the power consumed for each of our schemes in Figure 5. When we apply the LSA scheme, we find that, on an average, we save 16% power as compared to the case when no scheme is applied. Next, instead of performing link rate adaptation to the lowest value permitted by the traffic, we monitor the utilization of each link to ensure that it is always less than 70% utilized. [2] In case the utilization is above 70%, the link capacity is set to the next higher capacity level

---

[2] 70% is a rule-of-thumb value used by network administrators to keep the link delay at acceptable levels. We experimented with other values as well and obtained differing amount of power savings.

supported by the switch port. Also, unused links, instead of being completely turned off, are set at 10 Mbps to increase the availability of the system. As seen in Figure 5, the SL-aware LSA scheme takes slightly more power, while the availability and performance increase significantly (Table II).

Analyzing the links that have been adapted, we find that a majority (90%) of the links connecting servers to rack switches see light traffic and can be set at 10 or 100 Mbps for the entire duration of 5 days. Under 5% of the server to rack switch links see heavy traffic ($>$ 100 Mbps) and need to be maintained at the 1 Gbps rate. Next, we analyze the links between rack switches and tier-2 switches. In Figure 3, we plot the distribution of all links between rack and tier-2 switches that are rate adapted using our LSA scheme. As expected, less than 2% of the links that connect rack and tier-2 switches can be completely turned off; the majority of these links are set at 100 Mbps.

We simulate the effects of Network Traffic Consolidation (NTC) scheme and find the energy consumption to be significantly lower than with LSA scheme, though there is no redundancy with this option. The SL-aware NTC scheme, while consuming almost double the power of the baseline NTC scheme, offers a much higher performance and availability.

Next, we simulate the effects of the SLC scheme. Server load consolidation comes with its own limitations with respect to CPU, available memory, etc. During the entire trace duration, memory is the only limitation; other resources such as CPU and network bandwidth are never the bottleneck. For servers that currently have either 4 or 8 GB RAM, we consider the hypothetical case of their RAM being increased to 16 GB and thus load from 2 (or 3) servers maybe assigned to just one server. Using this server load consolidation combined with network traffic consolidation scheme yields the most benefits in terms of energy savings. The network energy consumed in this case is, on an average, 25% of the energy consumed by the baseline case. We plot the results from these schemes in Figure 5. In this paper, we ignore the energy saved by putting unused servers to sleep; if we include the potential savings from putting unused servers to sleep in the SLC scheme, the overall data center energy savings increase by an even bigger margin.

As described earlier, the energy savings are achieved at the cost of system reliability and performance. We use *the total number of independent paths between various servers* as a metric for evaluating the impact of various schemes on the reliability of the system. Our data center topology has 2 independent paths between any two servers residing on the same rack. Similarly, there are 4 independent paths between the servers on different racks and each server has 8 independent paths to reach the root switches in order to connect to systems outside the data center. In terms of performance, the latency experienced by packets can be impacted if the link utilization is high. Table II shows the tradeoffs between energy consumption, reliability and performance. It is clear that high availability and low energy consumption are conflicting goals - schemes with low energy consumption impact performance and high availability schemes consume more energy. There is a significant increase in the energy consumed by the SL-aware versions of NTC and SLC schemes due to the fact that the switches providing the redundant paths need to be turned on. The increase in the energy in these cases can be controlled by using lower power (though lower throughput) switches for backup paths.

Scheme	Average power (W)	Latency Impact?	Independent paths (intra-rack, intra-data center, outside data center)
No change	5402	no	2,4,8
LSA	4506	yes	traffic dependent
SL-aware LSA	4518	no	2,4,8
NTC	2256	yes	1,1,1
SL-aware NTC	4517	no	2,4,8
SLC	1357	yes	1,1,1
SL-aware SLC	4512	no	2,4,8

TABLE II
COMPARING AVAILABILITY AND POWER CONSUMED FOR VARIOUS SCHEMES

V. DEPLOYMENT CONSIDERATIONS

In the previous section, we assume the existence of an *Oracle* that has perfect information about arriving traffic and explore the possible energy savings by resorting to topology control and workload placement. While not practical, such an analysis helps us quantify the energy we can hope to save under ideal circumstances. We now discuss how we can deploy our schemes in a real data center (or other single administration domain networks such as an enterprise).

Tracking traffic workload dynamics: The various power saving schemes suggested in this paper require a network power controller that uses information about traffic on various links to compute the energy-efficient topology. Link utilization statistics can be collected from individual switches using SNMP. The topology control actions need to closely follow the changes in traffic conditions for greater power-efficiency without sacrificing performance. A simple watermarking based approach for adapting link capacities is one realtime deployment option. For instance, if a link's utilization crosses the high watermark of, say 70%, the link's linespeed can be adapted to the next level. This approach is similar to the service level awareness that we discuss in the previous section.

Another approach is to predict the traffic generated by various servers and the incoming traffic. Researchers have been looking at traffic prediction for a while and have developed many sophisticated prediction techniques. While prediction schemes are dependent on the nature of the traffic, for traffic with strong diurnal patterns such as ours, simple prediction models can be sufficient. We experiment with an AR(1) model to predict traffic to and from each of the 292 servers. Traffic traces from the first day are used to train the model; we then use the resulting model to predict traffic for the entire trace duration. Using this predicted traffic as input, we simulate all of our schemes to perform the required topology adaptation and workload consolidation. We compare the results to that of the *Oracle* approach. Unlike the ideal *Oracle* driven simulations in the previous section, the prediction driven approach can both over-estimate and under-estimate the linespeed settings of each link. Over-estimating a link's capacity implies we have set the link's linespeed at a higher level, when a lower rate would have sufficed, leading to a higher energy cost. Under-estimating, on the other hand implies incoming traffic might be more than the link's linespeed and sometimes might lead to packet loss. In Figure 6, we show the percentage of links whose capacities were over-estimated and under-estimated, when we simulate the NTC scheme on the predicted traffic. The average overestimation and underestimation of link capacities was 1.9% and 0.7% respectively for the NTC scheme for the entire trace duration. The power consumed by the prediction driven approach is close to that of the *Oracle* driven approach without incurring significant packet loss. The average loss across all links in the NTC scheme is shown in Figure 6 on the secondary y-axis. When simulating the effects of the SL-aware NTC scheme, our prediction model compares favorably with the *Oracle* approach in terms of energy saved and does not incur any packet loss. We experiment with other schemes as well for comparable results.

Mitigating transition performance impact: Our experiments show that the transition time for adapting linkspeeds is between 1-3 seconds. Such a large transition time can result in significant disruption of traffic. We plan to explore mechanisms such as buffering or ensuring the existence of backup paths for mitigating the performance impact during the transition period.

VI. RELATED WORK

Gupta *et al.* were amongst the earliest researchers to advocate conserving energy in networks [10]. Other researchers have proposed techniques such as putting idle sub-components (line cards, *etc.*) to sleep [7], [8], [10], [11], as well as adapting the rate at which switches forward packets depending on the traffic [7], [9]–[11]. Nedevschi [12] *et al.* discuss the benefits and deployment models of a network proxy that would allow end-hosts to sleep while the proxy keeps the network connection alive.

Chabarek [13] *et al.* enumerate the power demands of two widely used Cisco routers; further the authors use mixed integer optimization techniques to determine the optimal configuration at each router in their sample network for a given traffic matrix. Chen [14] *et al.* consider connection-intensive Internet services and propose load dispatching algorithms to reduce energy consumption in servers. While one of our power saving algorithms focuses on job allocation, we perform this operation from the point of view of saving power at network devices and we show considerable energy savings can be achieved when we combine server load with network load consolidation. Nedevschi *et al.* [8] propose shaping the traffic into small bursts at edge routers to facilitate sleeping and rate adaptation. Their research is complementary to ours. Further their work addresses edge routers in the Internet while our algorithms are for data centers and enterprise networks.

VII. CONCLUSION

Energy efficiency has become a high priority objective in most IT operational environments. Networks (including data center and enterprise networks) constitute an important part of the IT infrastructure and consume significant amounts of energy. Relatively little attention has been paid to improving the energy efficiency of networks thus far. Towards this end, in this paper, we make several contributions - (1) We propose algorithms for network power savings, based on the findings from our measurement study (2) We perform a case study of applying these algorithms on a real data center topology and a real Web 2.0 workload using power profiles of real switches and (3) We quantify the tradeoffs of our algorithms with respect to performance and reliability. Of our three schemes, the Server Load Consolidation (SLC) scheme performs the best with close to 75% energy savings. A more intelligent traffic routing using our Network Traffic Consolidation (NTC) scheme also yields significant network energy savings. We also incorporate service level awareness in our algorithms and show how network performance and redundancy can be traded off for energy savings. The tradeoffs provide an interesting knob for network operators to price the SLAs they offer to their customers. Customers who need more 9s of reliability can be charged more and vice versa. Our future work entails building a network power manager based on our findings and deploying it in a production network.

REFERENCES

[1] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings Of ISCA*, June 2007.

[2] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle, "Managing energy and server resources in hosting centers," in *Proceedings Of the 18th Symposium on Operating Systems Principles (SOSP)*, October 2001.

[3] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No Power Struggles: A Unified Multi-level Power Management Architecture for the Data Center," in *Proceedings Of ASPLOS*, March 2008.

[4] A. Greenberg, J. Hamilton, D. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," in *ACM SIGCOMM CCR: Editorial note*, January 2009.

[5] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," in *IEEE Computer*, December 2007.

[6] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A Power Benchmarking Framework for Network Devices," in *Proceedings of IFIP Networking (to appear)*, May 2009.

[7] M. Gupta and S. Singh, "Energy conservation with low power modes in Ethernet LAN environments," in *Proceedings Of IEEE INFOCOM (MiniSymposium)*, May 2007.

[8] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing Network Energy Consumption via Rate-Adaptation and Sleeping," in *Proceedings Of NSDI*, April 2008.

[9] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR)," *IEEE Transactions on Computers*, vol. 57, pp. 448–461, April 2008.

[10] M. Gupta and S. Singh, "Greening of the Internet," in *Proceedings Of ACM SIGCOMM*, August 2003.

[11] M. Gupta, S. Grover, and S. Singh, "A Feasibility Study for Power Management in LAN Switches," in *Proceedings Of IEEE ICNP*, October 2004.

[12] S. Nedevschi, J. Chandrashenkar, B. Nordman, S. Ratnasamy, and N. Taft, "Skilled in the Art of Being Idle: Reducing Energy Waste in Networked Systems," in *Proceedings Of NSDI*, April 2009.

[13] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *Proceedings Of INFOCOM*, April 2008.

[14] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in *Proceedings Of NSDI*, April 2008.