# Energy Conservation in Multi-Tenant Networks through Power Virtualization

Srini Seetharaman
Deutsche Telekom R&D Lab, Los Altos, CA USA
srini.seetharaman@telekom.com

## ABSTRACT

In the service-centric Internet, multiple virtual services (tenants) are overlayed on top of the same infrastructure (both in wide-area networks and in datacenter networks). We propose conserving energy, in this setting, by virtualizing network power consumed by each tenant, feeding back that information to the tenant, and incentivizing the tenant to conserve energy by making their bill proportional to this *virtual power*. However, virtualizing power in these multi-tenant networks is tricky since the network is not energy-proportional, i.e., the energy consumption and its monetary expenditure do not reduce with a decrease in load per component. We overcome this limitation by proposing a simple heuristic for billing, that further motivates tenants to align their workload in a manner conducive to optimization by the infrastructure provider.

## 1. INTRODUCTION

The Internet is rapidly becoming service-centric, where many of the services are being deployed over dedicated hardware that consume large amounts of energy; The total energy consumed by datacenter networks in the US was 3 billion kWh [30] in 2006. This figure is growing rapidly, owing to the increasing demand for cloud computing and a quest for competitive edge.

Most present-day efforts (Refer to §4 for more specific proposals) on conserving energy focus on servers and cooling, which account for about 70% of a datacenter's total power budget. However, there has been little effort towards conserving energy consumed by the networking elements; In datacenters, the network consumes 10-20% of its total power [11]. The strategies used in server environments cannot be easily applied for the networking hardware, because they are not *energy proportional* [3, 22], which means that the switch consumes a large amount of power (nearly 70-80% of the total) even in idle state, and they do not support different power states. The most effective approach in this domain, thus, involves powering down unused switches and links [2], for a given workload.

However, having no control on the workload prevents the network from achieving the best possible energy saving[1]. This is especially true in *multi-tenant networks* of today, where there are multiple virtual services (or tenants) hosted as an overlay over the infrastructure (or substrate). Example of a tenant service includes an instance of Hadoop running in a cluster of workstations [14] or a content-distribution service in the wide-area network [1]. In these environments, there are two problems: 1) the tenant is oblivious to its energy usage and configures workload solely based on performance, and 2) commercial systems (e.g., EC2 [6]) charge a flat-rate fee based on absolute number of bytes transferred and not based on duration or type of network activity. These two issues can lead to wasteful behavior by the tenant, from the perspective of energy consumption.

To address this problem, we propose *virtualizing network power in such multi-tenant networks and incentivizing the tenants to conserve power*, so that the tenants might reshape their activity to minimize their energy footprint. By charging each tenant for their exact contribution to the overall infrastructure power, the consequent system will have much reduced power. The tenants may either curtail their usage and reduce the overall load, or align workload in a manner conducive to further power optimization at the substrate level. Note that any saving is possible *only when the substrate layer is capable of routing a given workload, which has been* adjusted for reduced expense by the tenant, and then powering-down unused elements (switches/links). Our work, thus, builds over our earlier proposal of energy-aware network-layer topology control [2] to identify ways to motivate a tenant to becoming energy-conscious.

In this paper, we describe how to make this proposal a reality and get tenants to cooperatively re-adjust their workload. This paper presents our preliminary ideas and is a precussor to a full-fledged analysis. Our approach is equally applicable to both wide-area overlay (virtual) networks and datacenter networks, with the common criteria being that the infrastructure provider is hosting many higher-level tenants (service providers).

The rest of the paper is organized as follows: §2 gives more details on the technical challenges involved in virtualizing power. §3 discusses consequence of virtualizing and making the bill proportional to the energy usage. We present related work in §4. We conclude the paper and provide future work in §5.

---

[1] We use the terms power and energy interchangeably, unless otherwise mentioned.

# 2. VIRTUALIZING POWER

As mentioned in the previous section, there are two high level actions proposed in this paper:

- Virtualizing network power consumed by each tenant and feeding back that information to them.
- Incentivizing the tenant to conserve power by making their bill related to this virtual power.

This section describes the problem domain and the technical challenges associated with implementing the proposals.

## 2.1 Model

To better understand the network power, we adopt the empirically-derived power model from [18]:

$$\text{Power}_{\text{switch}} = \text{Power}_{\text{chassis}} + \#\text{ cards} \times \text{Power}_{\text{linecard}} + \sum_{i=0}^{\#\text{ ports}} \text{Power}_i(\text{rate}_i) * \text{util}_i,$$

where the $\text{Power}_{\text{chassis}}$ dominates the overall value. [22] presents power measurements for different switch models for different levels of usage. We observe, from Figure 5 of [22] and Table 1 of [2], that a switch with all ports down consumes 94.8% of the power it typically consumes when all ports are up and fully-utilized. This is partly due to fixed overheads such as fans, switch chips, and transceivers that waste power even at low loads. This illustrates the energy disproportionality.

Figure 1 illustrates two virtual networks (tenants) coexisting over the same infrastructure[2]. Each tenant has different traffic load, as well as different power requirements. The infrastructure, on the other hand, serves as a union of the load (both power and traffic) of the hosted tenants. If network elements were energy-proportional, then virtualizing power is straightforward; the individual *virtual power* will be in direct proportion to the load. This is the case illustrated in Figure 1. In reality, however, lack of energy-proportionality makes all powered-on devices to be operating at (almost) its highest power level. In this latter case, virtualizing power is not straight-forward and we need special heuristics to achieve that (The next subsection presents one such heuristic).

Once the virtual power is computed and conveyed back to the tenant, the tenant (duly incentivized to conserve expenses) will adjust its network activity: either migrate virtual machines to consolidate load, or migrate existing/incoming requests to different replicas of the resource, or reroute their own traffic such that the active set of virtual links represent the least possible expenditure for the tenant. The exact operation will dependent on the level of cross-layer awareness it has; It may be at three different levels:

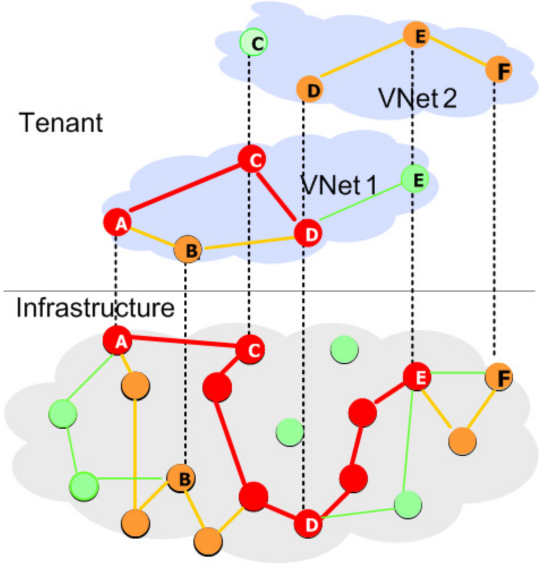

**Figure 1: Typical scenario in a multi-tenant network. The colors represent the load (with thick/ red being highest and thin/ green being lowest).**

- The tenant is only aware of end-hosts and the substrate takes care of routing traffic between each pair of hosts. This is common with most server applications today.
- The tenant possesses knowledge of the elements comprising the inter-host path, either by inferring using measurement tools (like `traceroute`) or by querying the infrastructure operator.
- The tenant is capable of routing traffic belonging to them, with the substrate applying necessary restraints to insure strict isolation (of bandwidth, policy, routing decisions, privacy). This is akin to the FlowVisor [26] slicing in OpenFlow [21] networks.

Based on the amount of participation each tenant has in its network management, the exact entity optimizing the routes for low power consumption will vary.

Note that the virtualizing model assumes a fair (across tenants) and honest (in routing and billing) infrastructure layer that is ready to power-down network elements when the workload allowed it.

## 2.2 Heuristics for Virtualizing Power

As mentioned above, the lack of energy proportionality makes it hard to identify the contribution of each tenant to the power consumed by a switch, linecard or link. We propose the following simple virtualization

---

[2] The link at the tenant layer is referred to as a "virtual link". Each virtual link is comprised of multiple physical switches and links.

model:

$$\text{Virtual power}_{\text{element i}} = \frac{\text{Power}_i}{\#\text{ sharing tenants}}$$

$$\text{Power}_{\text{virtual link j}} = \sum_{\text{element i} \in \text{j}} \text{Virtual power}_i$$

The above equations suggest that we share the energy used per element (link or switch) among all the tenants using that element. Note that no tenant has direct visibility into what elements other tenants are using. Based on the level of cross-layer awareness available, each tenant either sees a list of elements and the corresponding bill for each, or just a cumulative value per virtual link. Both suffice in allowing the tenant to re-adjust the workload.

Though simplistic, the aforementioned virtual power model is powerful because it penalizes tenants that do not share resources. Since the power bill is not load-dependent, the tenant is further incentivized to send as much of its traffic on the same (already purchased) resources, rather than invest in new (possibly powered-down) resources.

## 2.3 Monetization of Virtual Power

As shown by several pieces of research in mechanism design, payments are the easiest and fool-proof way of incentivizing. In the first method, the infrastructure provider passes on the expenses for the power directly to the tenant, without double billing the tenants or billing for unused resources. Thus, the virtual power bill, which is proportional to the virtual power, of all tenants add upto the total energy-related operational expenses of the infrastructure provider.

Alternatively, the infrastructure provider may treat the overall energy as a resource and sets up a resource management framework around it. Then, the infrastructure provider can adopt an economic approach to auction these resources to the individual tenants. Based on how much power a tenant purchased over a particular duration, the infrastructure provider will provision the flows of that tenant. This is similar, in a way, to the approach in Muse [16], which improves energy efficiency under fluctuating load by allowing individual services to "bid" for resources and maximizing the network productivity. This form of *energy budgeting* is promising and reserved for further study.

A third way of allocating power resources is by allocating a fixed number of initial credits for a fixed fee, depleting the energy as a *capacitor*, and then gradually replenishing credits over time. The Cinder operating system [27] uses this capacitor model to schedule processes based on their pre-profiled energy footprint. We reserve this for future study as well.

In all three ways a tenant can purchase more power and possibly achieve a better QoS. There is no reason or means to block that.

## 2.4 Strawman Implementation

There is considerable discussion underway at the IETF Operations and Management Area Working Group (opsawg) on defining a Power Monitoring Architecture [23]. The general idea is to have hierarchies of *power monitors* (possibly implemented within a switch), with each of them performing some level of summarization. To virtualize power consumption among tenants, we propose building a specialized top-level power monitor parent, called *PowerVisor*[3], that is capable of virtualizing the power states. Each tenant, then, subscribes to this PowerVisor, over standardized configuration protocols, to obtain updates in a periodic manner or when state changes. This design is similar to that adopted by the FlowVisor in virtualizing routing control in OpenFlow networks [26].

# 3. CROSS-LAYER AND CROSS-TENANT INTERACTION

## 3.1 Cross-layer Interaction

As mentioned in Section 1, the substrate layer must implement some form of intelligence in powering down elements, once the tenants reshape their workload. Our earlier prototype, called ElasticTree [2], goes even beyond performing simple routing and powering down unnecessary elements; ElasticTree, in datacenter scenarios, bin-packs all given traffic demands within a minimal set of elements and reroutes traffic around using OpenFlow, thereby allowing the network operator to power down additional unused elements. Figure 2 illustrates the outcome of ElasticTree bin-packing. By reserving a small margin of resources for unexpected demand bursts, ElasticTree avoids potential congestion, while reducing the overall network power consumption.

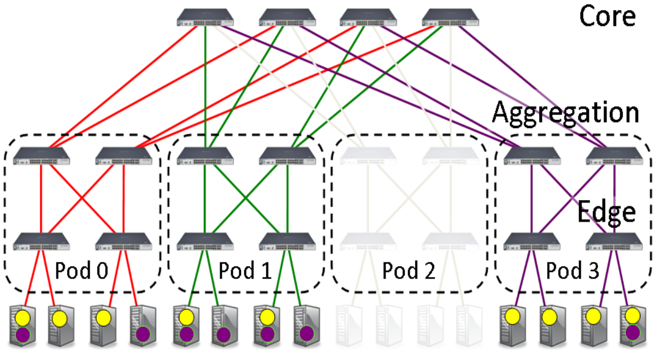

**Figure 2: Illustration of 2 tenant applications in a datacenter network. The circles of same color represent the virtual machines working together in a cluster. The grayed-out elements are powered-down.**

---

[3] The name is inspired by the HyperVisor in the server virtualization domain.

If the substrate layer is capable of offering other low power states in the future (e.g., exact energy proportionality), then the substrate layer can perform the necessary mapping of tenants' workload to different states of the physical hardware.

### 3.2 Cross-tenant Interaction

Virtualizing power across tenants and billing each based on individual usage creates interesting interactions. It is worthwhile exploring some of them in order to facilitate a win-win situation.

*Load-based conflict.*

Since the cost model does not take into account the load of each tenant, it is possible that one tenant overloads certain links and switches, forcing other tenants off those elements. This can lead to a starvation of some tenants, though they have the budget for powering on those links/switches. Furthermore, some malicious tenants can deliberately overload parts of the network that they do not need. The infrastructure has to take necessary precautions to avoid this.

*Collusion or Masquerading.*

Since the virtual power is computed based on the number of tenants sharing an infrastructure, it is foreseeable that two (or more) tenants may collude to create a single super-tenant. In such cases, the infrastructure provider may be unable to distinguish between individual tenants and bill the colluding tenants lower than the other tenants (For instance, if there are a total of 5 tenants, two of them can collude to reduce the price from $2/5$ to $1/4$ of the total). Similarly, a single tenant can masquerade as two different tenants that send traffic between one another and affect the cost sharing equation. Thus, implementing fairness in billing needs special consideration. We reserve that for future study.

*Energy Trading.*

Our budgeting model, as presented in Section 2.3, is conducive to creating an energy-trading market in the tenant level as well. Different tenants may purchase energy credits in bulk and trade with other tenants, e.g., tenant $A$ purchases $X$ credits from tenant $B$, and tenant $B$ negotiates with the infrastructure provider in keeping alive certain links/switches that tenant $A$ needs. This trading needs to be further analyzed for consistency.

## 4. RELATED WORK

Previous proposals on server power management include using better components (low-power CPUs [9], power supplies, and water-cooling), smart cooling [8], dynamic voltage and frequency scaling of CPUs [7], energy-conscious provisioning [16] and consolidation [29], taking powernaps [19], and energy-aware server provisioning and load dispatching [10]. All previous work in this space has solely focused on computing components and associated energy saving, while ignoring the repercussions on the network.

There exist previous work on expense-conscious workload migration, across different datacenter locations, without impacting server capacity [24, 25]. The granularity of the load migrated depends on the energy-proportionality of the datacenter, with higher proportionality allowing finer granularity. All these efforts are orthogonal to our approach, and are more concerned about improving the power bill rather than conserving the absolute energy consumed.

Existing approach in conserving network energy propose powering down unused switches and ports [2], link rate adaptation [4, 18], reshaping traffic into bursts [28], or adjusting switch configurations [5, 18]. Other efforts that propose putting network components to sleep [12, 13] will be enabled once the IEEE 802.3az Task Force proposals [15] hit the market. However, neither of these focus on allowing the applications to reshape their own workload.

There exist research efforts in identifying the power consumed by individual applications in operation systems [27], and by clusters of virtual machine [20]. However, ours is the first to focus on virtualizing network power consumed by individual network services.

## 5. DISCUSSION AND FUTURE WORK

The underlying premise of this paper is that making the hosting charges incurred by a tenant proportional to the energy consumed by that tenant will incentivize the tenant to conserve usage and align workload in a manner conducive to further power optimization at the substrate level.

Our paper is the first to propose virtualizing the network power of each tenant. In a way, this approach takes the network virtualization principle one step closer towards complete isolation of virtual networks. Note that existing proposals for network virtualization (e.g., FlowVisor in OpenFlow networks [26], or Virtual LANs) only provide isolation of bandwidth, topology and routing control isolation. Thus, adding power to that list is a crucial step.

*Switch vs Servers.*

From power measurements in the Powernet project [17], we learn that the power wattage of switches are comparable to that of desktop and monitors. However, desktops and monitors, unlike switches, are more energy proportional and have multiple energy states. This emphasizes the need to focus on the network energy consumption. In the future, when the network becomes more energy-proportional or, at the least, provides different operating states, we may be able to completely outsource power management to individual tenants, while maintaining/enforcing some global policies. This is similar in spirit to the idea presented in [20].

### *Heat Consequence and applicability to cooling.*

As pointed out by authors of [16], concentrating load on fewer devices and powering down unused elements reduces the heat output as well, thereby reducing the energy consumed for cooling.

In a manner similar to virtual network power, it is foreseeable that feeding back geographical locality information to the tenant can help power down cooling devices. In that case, the important factors in play are:

- Number of elements deployed and overall power consumed by them
- Virtualizability (time, space, tenant) of the power
- Quantization level or power states of the devices

### *Future Work.*

The important next steps are to construct a realistic large-scale multi-tenant network (with possibly Hadoop [14]-like tenant applications) for evaluating the different dynamics. The network should virtualize power using the three different charge models listed in Section 2.3. We plan to build this system atop our OpenFlow-based ElasticTree prototype as it is has all the instrumentation for the necessary operations (power measurement, virtualization of flows, and topology control).

Additionally, we wish to address the following open questions in our work:

- *What if tenant utility is a factor?* Our simple heuristic ignores utility. The previous work in the computing space investigate the tradeoff between service demand, operational expense and the user experience (utility) [10, 16]. Similar to their approach, we may be able to include utility in the charge equation of Section 2.3.
- *What other non-payment incentives are possible to reduce energy consumption?* We plan to investigate using network bandwidth, oversubscription, or other QoS guarantees as an incentive to tenants, such that better performance is provided to the tenant only if their virtual power is kept low.
- *How does infrastructure and tenant heterogeneity affect power virtualization?* We believe the cross-tenant interaction will, then, be more significant.
- *How to achieve an energy-aware tenant design?* This will insure that the virtual network is proactively designed for power conservation, thereby improving the tradeoff between performance and power. Locality will feature as a key parameter in this design.

## Acknowledgements

We thank Ramesh Johari and Brandon Heller for inspiring discussions that laid the groundwork for this paper. We also thank Andreas Gladisch, Rob Sherwood, and the anonymous reviewers for their comments on the work.

## 6. REFERENCES

[1] Akamai. `http://www.akamai.com`.

[2] B. Heller et al. ElasticTree: Saving Energy in Data Center Networks. In *Proceedings of USENIX NSDI*, April 2010.

[3] L. A. Barroso and U. Hlzle. The case for energy-proportional computing. *Computer*, 40(12), 2007.

[4] C. Gunaratne et al. Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR). *IEEE Transactions on Computers*, 57:448–461, April 2008.

[5] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power Awareness in Network Design and Routing. In *IEEE INFOCOM*, April 2008.

[6] Amazon Elastic Compute Cloud (Amazon EC2). `http://aws.amazon.com/ec2`.

[7] E. Elnozahy, M. Kistler, and R. Rajamony. Energy-efficient server clusters. In *In Proceedings of the 2nd Workshop on Power-Aware Computing Systems*, pages 179–196, 2002.

[8] C. P. et al. Smart Cooling of data Centers. In *Proceedings of InterPack*, July 2003.

[9] D. G. et al. Policies for dynamic clock scheduling. In *USENIX OSDI*, 2000.

[10] G. Chen et al. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *Proceedings of USENIX NSDI*, pages 337–350, 2008.

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

[12] M. Gupta and S. Singh. Greening of the internet. In *ACM SIGCOMM*, pages 19–26, 2003.

[13] M. Gupta and S. Singh. Using Low-Power Modes for Energy Conservation in Ethernet LANs. In *IEEE INFOCOM*, May 2007.

[14] Hadoop. `http://hadoop.apache.org/`.

[15] IEEE 802.3az. `ieee802.org/3/az/public/index.html`.

[16] J. Chase et al. Managing energy and server resources in hosting centers. In *Proceedings of 18th ACM SOSP*, 2001.

[17] M. A. Kazandjieva, B. Heller, P. Levis, and C. Kozyrakis. Energy dumpster diving. In *Proceedings of HotPower*, 2009.

[18] P. Mahadevan, S. Banerjee, and P. Sharma. Energy proportionality of an enterprise network. In *1st ACM SIGCOMM Workshop on Green Networking*, 2010.

[19] D. Meisner, B. T. Gold, and T. F. Wenisch. Powernap: eliminating server idle power. In *Proceedings of the 14th ASPLOS*, pages 205–216, 2009.

[20] R. Nathuji and K. Schwan. Virtualpower: coordinated power management in virtualized enterprise systems. *SIGOPS Oper. Syst. Rev.*, 41(6):265–278, 2007.

[21] The openflow switch. `http://www.openflowswitch.org`.

[22] P. Mahadevan et al. A Power Benchmarking Framework for Network Devices. In *Proceedings of IFIP Networking*, 2009.

[23] Power Management Architecture. IETF Internet draft, work in progress, `http://tools.ietf.org/html/draft-claise-power-management-arch-00`, July 2010.

[24] More on Virtualizing Power - Cut Costs while Improving Reliability. `http://www.powerassure.com/about-power-assure/blogs/virtualizing-power`, 2009.

[25] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In *Proceedings of ACM SIGCOMM*, 2009.

[26] R. Sherwood et al. Carving Research Slices Out of Your Production Networks with OpenFlow. In *Proceedings of the ACM SIGCOMM Demo*, 2009.

[27] S. M. Rumble et al. Apprehending Joule thieves with Cinder. In *Proceedings of ACM MobiHeld*, Aug 2009.

[28] S. Nedevschi et al. Reducing network energy consumption via sleeping and rate-adaptation. In *Proceedings of the 5th USENIX NSDI*, pages 323–336, 2008.

[29] S. Srikantaiah, A. Kansal, and F. Zhao. Energy aware consolidation for cloud computing. In *Proceedings of USENIX HotPower workshop*, December 2008.

[30] U.S. Environmental Protection Agency. Data Center Report to Congress. `http://www.energystar.gov`.