16

Energy-Efficient Networking in Modern Data Centers

Dominique Dudkowski and Peer Hasselmeyer
NEC Europe Ltd., NEC Laboratories Europe, Heidelberg, Germany

16.1 Introduction

Achieving energy-efficient operation of information and communication (ICT) technology has become one of society's primary objectives, being actively investigated in research and development. It is estimated that 8–10% of global carbon dioxide (CO_2) emissions are because of the ICT industry (http://www.vertatique.com/ict-10-global-energy-consumption, http://europa.eu/rapid/press-release_IP-13-231_en.htm).

Data centers play an essential role in today's ICT infrastructure by serving as the backbone for many kinds of electronic services. They make up a significant part of ICT's total energy consumption, accounting for 23% of ICT's global CO_2 emissions (http://www.gartner.com/it/page.jsp?id=530912). It is assumed that in the United States all data centers combined consumed between 1.7% and 2.2% of the total US electricity consumption in 2010, while worldwide data center energy consumption was around 1.3% [1].

This amount is certainly significant, and data center operators and researchers are working hard to reduce data center energy consumption. Approaches toward increasing data center energy efficiency mainly target computing, storage, the network, and cooling infrastructure individually or in a joint way, as illustrated in Figure 16.1. Computing nodes, including high-end, mid-range, and volume servers, together may consume as much as 40% of a data center's total energy [2]. Accordingly, computing has received the most attention by researchers and developers. In contrast, the typical energy consumption of a data center network may range from 5% [2] to 12% of overall data center consumption when servers are fully loaded [3].

Due to the small fraction of the overall energy consumption that is taken up by the network, it is necessary to motivate the need for energy-efficient solutions in the network. The most relevant arguments are:

Green Communications: Principles, Concepts and Practice, First Edition.
Edited by Konstantinos Samdanis, Peter Rost, Andreas Maeder, Michela Meo and Christos Verikoukis.

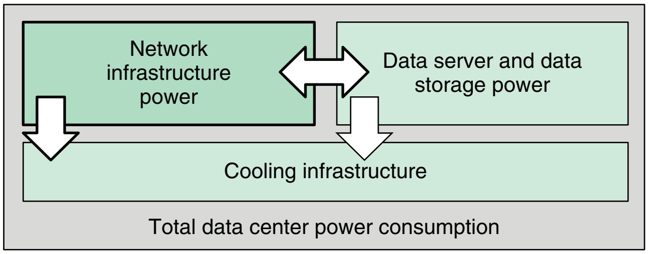

Figure 16.1 Energy efficiency considerations in the data center

16.1.1 Energy-Proportional Computing

Research on the computing side of the data center has been focused primarily on making individual servers and complete server farms *energy proportional* [4, 5], meaning that their energy consumption depends linearly on their workload. The same concept can be applied to the network, meaning that the energy consumption of network links, network elements, and the network as a whole depend linearly on their utilization (in terms of fraction of the maximum bandwidth provided).

Given that the server side is managed energy proportionally, the network's relative share of overall energy consumption increases with a decrease in computing load on the data center. Abts et al. [3] illustrate this under the premise of energy proportionality implemented on the computing side, but not on the network side. While 12% of the DC energy consumption can be attributed to the network when the servers are fully loaded, the network's share increases to 50% when the data center (both servers and network) is utilized only 15%.

16.1.2 Boost in Link Bandwidth

In data center networks, link bandwidth is constantly increasing. Common link rates are currently 1–40 Gbps, with different bandwidths typically used in different tiers of a data center network, for example, 10 Gbps links in the core and aggregation tiers, and 1 Gbps links in the access tier [6]. Reviriego et al. [7] assume that 1 Gbps Ethernet links consume 1 W while 10 Gbps Ethernet links reach 5 W. Although efficiency (in bps/W) is increasing with every generation of networking equipment and absolute power requirements are decreasing for the same bandwidth, with growing bandwidth, for example, from 10 Gbps to 100 Gbps, the power consumption still increases fivefold [8]. Because a data center network is composed of a large number of elements, changing link bandwidth results in a large increase in power consumption and therefore deserves consideration.

16.1.3 Impact on Cooling Infrastructure

Besides the direct benefits from saving energy in the network, less energy consumed by network elements decreases the requirements on cooling (Figure 16.1). While this fact is recognized on the server side and, for example, considered by Ahmad and Vijaykumar [5], it also holds for networking equipment and can make a big difference. Only when all servers *and* switches in the same rack are deactivated can the cooling auxiliaries be shut down in that segment too, reducing energy consumption accordingly.

16.1.4 Impact on Power Distribution Infrastructure

Similar to the cooling infrastructure, a second indirect impact is on the power distribution auxiliary infrastructure. For example, Pelley et al. [9] note that it is very unlikely that all servers in a network peak at the same time. A time series of 5000 servers is also given by Barroso and Hölzle [4], noting that peak power is achieved only in very few cases. Furthermore, according to Pelley et al., statistical variation of usage is high. These reasons have led to power distribution infrastructures being highly oversized due to conservative measures. While this does not primarily mean OPEX-side problems, this time it is capital expenditures on power distribution equipment that makes up a significant part of overall data center CAPEX (e.g., $10–$100 millions, according to Pelley et al. [9]).

The same reasoning applies to data center networks, as, intuitively, network traffic also correlates with computing load. Applying energy control on the network side therefore leads to similar effects in reducing power distribution needs, translating into additional CAPEX savings.

While these are strong arguments for considering energy saving in the network, current practices in operational data centers are still limited to solutions on the computing side, while the complete network remains operating 24/7. One reason for this disparity is that enabling technologies are readily available on the computing side, in particular, virtualization and virtual machine migration enabling freeing and shutting down physical machines. On the network side, however, enabling technologies for flexible traffic management are not yet available in production data center networks, although appearing in recent research, for example, OpenFlow (http://www.openflow.org).

16.2 Energy Efficiency in Data Center Networks

Work on energy-efficient networks is amply available in the literature, some of which is applicable to any type of fixed network, and some being specific to data centers. The main consensus is that energy savings should never occur at the expense of performance. The main techniques are described in this section.

16.2.1 Dynamic Link Rate Adaptation

With the increase in link bandwidth, it makes sense to allow links to run at different speeds, as the maximum link bandwidth is usually not always needed. Such scaling is supported by current technology, such as InfiniBand, which can operate at single, double, and quad data rates ranging from 2.5 to 40 Gbps [3]. Such variation in link bandwidth is sometimes classified as *performance state*, in contrast to power state, such as sleep mode [2].

If energy proportionality is given, link rate adaptation influences energy consumption according to link speed. As a network is often loaded only lightly, link rate adaptation bears large potential for energy savings. Abts et al. [3], who apply link rate adaptation in their approach, indicate a strong benefit of scaling link bandwidth: 73% of the time they are able to operate a link at only 2.5 Gbps, while it runs at 40 Gbps only 5% of the time. Gunaratne et al. [10] similarly report that a link can be run at a lower data rate than its maximum for more than 80% of the time. Even better energy proportionality can be achieved by using traffic bursts with long intermittent sleep phases [7].

16.2.2 Link and Switch Sleep Modes

Changing the operational mode (also called the *power state*, in contrast to performance state [2]) of network elements from active to a lower power state such as “idle” or “off” is another promising approach. Controlling the state of network elements is possible at the level of individual links and/or complete switches, a choice that depends on the capabilities of the network element. While link rate adaptation is suitable for elements that operate near energy proportionality, network elements are often not energy proportional and according to Heller et al. [11] consume just 8% additional energy when fully loaded compared to the idle state. In that case, turning links and/or switches on and off makes more sense, which Heller et al. consider in the proposed ElasticTree.

In order to shut down and start up network elements and individual links the data center needs to offer appropriate capabilities under programmatic control. As switches at this time do not generally support a suitable shutdown function, a viable alternative for controlling the power states of switches is via remotely switchable power strips. In case native shutdown functions are supported, reactivating a network element can be implemented via Wake-on-LAN or similar techniques. Controlling individual links of a network switch, on the other side, may be commonly achieved via SNMP.

16.2.3 Network Topology

The topology is probably the most critical parameter of a data center’s network because it determines various performance and operational limits, such as latency, redundancy, and maximum aggregate end-to-end bandwidth between computing nodes. Besides performance, different network topologies have different characteristics regarding the ability to apply energy saving strategies. Figure 16.2 provides an overview of data center network topologies from the literature that are considered valuable with respect to energy efficiency.

The common data center network topology as shown in Figure 16.2.a has large oversubscription ratios (i.e., the aggregate bandwidth between layers decreases toward the top of the tree) and is therefore inadequate for cloud-style traffic loads that exhibit unpredictable traffic patterns with a large fraction of traffic crossing rack boundaries. Fat trees, and in particular their incarnation as folded-Clos networks (Figure 16.2b), have been proposed as alternative topologies featuring full bisection bandwidth. Compared to large switches commonly used in hierarchical networks, fat tree networks built from multiple smaller off-the-shelf switches can save as much as 56.6% of energy according to Al Fares et al. [6] for comparable bandwidth and performance, even if all switches are on.

Because the fat tree provides much more flexibility in deactivating individual links and whole switches while connectivity can still be maintained, Heller et al. [11] propose the Elastic Tree, which is backed by a subset of the fat tree (e.g., configuration shown in Figure 16.2c). The authors propose to turn links and switches on and off, constantly adjusting the set of active

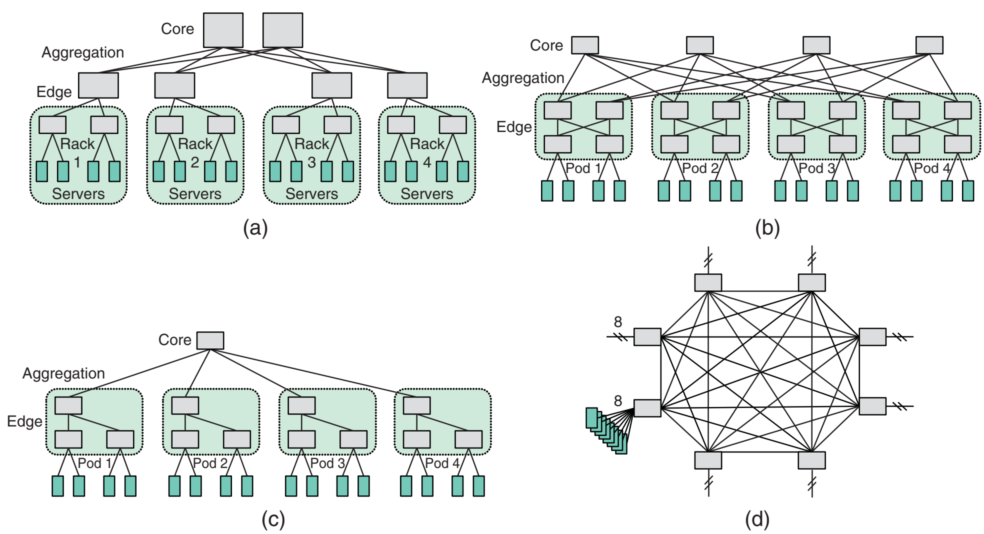

Figure 16.2 Network topologies considered in the context of energy efficiency. (a) Common date center network topology [6, 10]. (b) Fat tree topology, based on [6, 10]. (c) Elastic tree (fat tree subset), based on [10]. (d) Flattened butterfly topology for 64 nodes based on [3]

network elements according to network performance and fault tolerance requirements, leading to power savings of 25–40% compared to an always-on fat tree. If used in a large network, the approach exhibits energy-proportional behavior on the network level despite the individual elements not being energy proportional.

The butterfly topology proposed by Abts et al. [3] and shown in Figure 16.2d is claimed to be inherently even more energy-efficient than fat tree networks.

16.2.4 Combination of Approaches

Energy saving techniques for data center networks can be combined with one another relatively easily. The conceptual difference of performance states and power states according to Ref. [2] is exploited in an illustrative way by Nedevschi et al. [12]. The authors apply both sleeping and dynamic link rate adaptation by putting network elements to sleep while idle and by adapting network link rates according to traffic volume. The authors' approach is particularly interesting because it exhibits high network-wide energy proportionality on the network as a whole.

Another important aspect is how to combine network-centric energy saving techniques with computing and cooling management in the data center. Ahmad and Vijaykumar [5] explore the combination of computing and cooling in energy management (also illustrated in Figure 16.1), noting that only a joint approach can optimize the overall energy consumption of server and cooling power, instead of, for example, creating hotspots that lead to increased cooling cost.

16.2.5 Network Performance

Last but not least, maintaining data center performance is vital despite the attempts to save energy using any of the aforementioned techniques. The general observation [3, 11] is that a network operated close to its capacity may lead to packets being dropped or delayed (back-pressure) depending on the flow control mechanism implemented in the network. For instance, according to Ref. [3], for values above as low as 75% of link utilization in the case of the considered dynamic link rate adaptation, latency increases substantially, and the network is likely to saturate. An additional constraint to consider is the fact that changing the speed of a link involves reconfiguration latencies that are typically in the range between 100 nanoseconds and 100 microseconds [3]. In contrast, activating and deactivating links takes more time, because higher layers (memory, operating system) incur more overhead. Nedevschi et al. [12] assume transition times of up to 10 milliseconds.

For these reasons state-of-the-art approaches provide heuristics, where link rates are adapted on the basis of expected utilization with suitable thresholds (e.g., in Refs. [3] and [10]) to avoid saturation. In the case of sleep modes and the longer times involved to change between sleep and operational state, Heller et al. [11] provide heuristics, predictive models, and safety margins, such that links and network elements are turned on soon enough to provide sufficient capacity at any time. This is similar to server side approaches, for example, in Ref. [5], where a number of spare servers are provided in excess to currently used servers to be able to quickly handle additional load.

16.3 A Joint Energy Management Solution

Most work on energy efficiency has so far focused on particular aspects of single domains, in particular, servers or networks. It is nevertheless apparent that energy efficiency can be improved by looking at multiple domains together than at a single one in isolation. We therefore developed an approach to network energy management that performs traffic placement in the data center network using knowledge from the server and the network domains to achieve higher energy savings than would be possible by looking at the network domain alone. The main idea of our approach is to aggregate traffic in certain parts of the data center in order to increase the chances for shutting down switch ports and complete switches in the remaining parts of the data center, similar to the Elastic Tree [11]. The decisions for placing traffic are guided by utilization metrics from both the network and the server side.

Our approach rests on the notion of traffic flows. A flow is a sequence of related packets. Typical examples of flows are individual TCP connections, all traffic to a particular server, and traffic related to a particular service, for example, Web traffic. The granularity of flows can vary widely and is influenced by the traffic patterns exhibited by the ensemble of applications deployed in the data center. Our approach works with any flow definition, although its effectiveness may be influenced by traffic patterns.

All traffic flows need to be routed inside the data center network. Our approach is inspired by OpenFlow, but can also be realized with other technologies, such as SNMP and MPLS. OpenFlow is a protocol that makes routing control in the form of forwarding tables accessible to external controlling entities. Forwarding tables contain entries that specify which actions to perform to packets matching a given set of criteria. Criteria include information gathered from the Ethernet, IP, and TCP/UDP headers of data packets. Actions include rewriting certain fields of packet headers as well as forwarding instructions.

We assume that controllers decide on the routes that traffic must follow. The routes are established on demand when a new flow is encountered the first time. Once a route is determined, the flow is enforced by setting appropriate flow entries in OpenFlow-enabled switches that are on the route. Contrary to some existing approaches, like Elastic Tree, our approach does not actively reroute existing flows in the data center network. It rather restricts itself to influencing the placing of new flows on their creation, relying on a gradual accumulation of traffic in certain parts of the data center over time. As a consequence, and depending on network load, links and switches can eventually be freed from any traffic and shutdown. This approach is computationally inexpensive as computational effort for calculating energy-efficient routes is needed only at flow creation time.

Because our approach works on both switch and link level, it provides good energy proportionality, which our performance evaluation shows. Although we consider power states only according to Ref. [2], energy saving techniques that relate to performance states, such as dynamic link rate adaptation [3, 10], can be integrated easily. Moreover, our approach does not depend on a particular network topology, and besides the fat tree that we assume in this chapter, any other topology is also supported.

16.3.1 Description of Approach

The idea of our approach is to anticipate which network elements or individual ports are likely to become inactive in the near future. By inactive, we mean that a network element or port does not process any flow, that is, it does not forward any packets. Intuitively, the smaller the amount of flows processed by a network element, the easier it becomes to turn off the switch by either waiting for the remaining flows to time out or to be actively rerouted. In this chapter we consider timeouts only.

Figure 16.3 illustrates the overall approach in a small fat tree network that is composed of commodity network switches supporting OpenFlow. The switches are controlled by the management system that is responsible (among other things) for traffic assignment. The management system is enhanced by energy control functions as described here.

In order to characterize a network element in terms of its current utilization, a scalar utilization metric is calculated for each network element. The metric expresses the approximate utilization of the switch, for example, in terms of the number of flows the switch is processing. The utilization metric of a switch is recalculated whenever the load changes, for example, when a flow is created or removed. The management system retains a record of the utilization metrics of all the switches. The metrics are used later on to find optimal traffic paths.

As an example, switch S_1 in Figure 16.3 reports its flow information to the energy management module in the management system in step 1. The energy management module then recalculates the metric for switch S_1 in step 2 and stores the information for later use.

The main process is triggered by the arrival of an unknown flow. In Figure 16.3, switch S_7 receives a traffic flow for which it does not have routing information. In step 3, it hands the request over to the management system's interceptor.

In a typical data center environment, the request is then handed to the load balancer (step 4), which selects possible target hosts (virtual machines) for the request. For instance, the request may originate from a Web server application, which requests one instance of a database application, of which multiple instances are running on different hosts in the data center. In current setups, the load balancer outputs exactly one instance. We assume that the load balancer function is changed slightly to output a small set of potential targets in order to provide host diversity for increased flexibility in path selection.

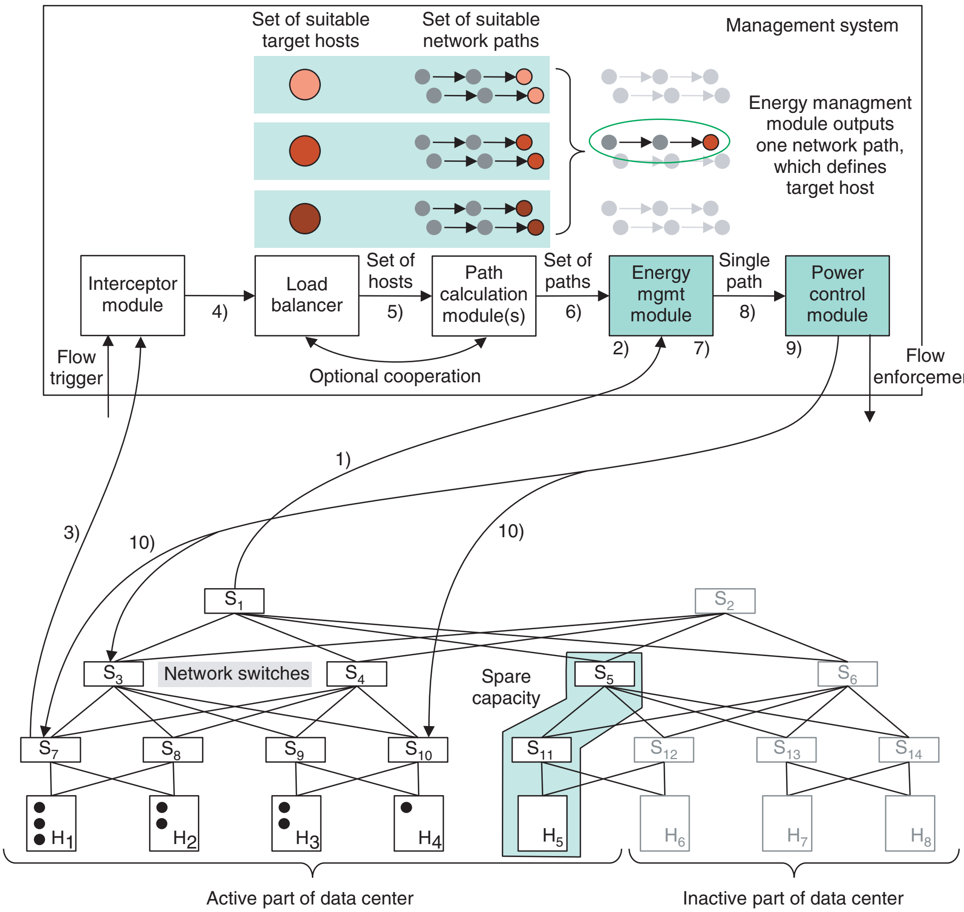

Figure 16.3 Overview of the proposed approach

In step 5, the set of hosts determined by the load balancer is handed to the path calculation module that computes paths under consideration of performance constraints such as bandwidth, latency. For each target host determined by the load balancer, this module calculates a set of suitable network paths, for example, all paths that have a minimum amount of bandwidth available to serve the request. This step results in multiple and diverse paths that are input to the energy management module in step 6.

Recall that the energy management module possesses, for any network element and port, a metric that characterizes the element and port in terms of utilization. The energy management module uses these values to select the path from the previously calculated ones, which it considers most suitable in terms of energy efficiency. Note that all paths at this point satisfy performance requirements, so the energy module can freely decide which one is the most suitable from its perspective.

Assuming network elements as the unit of granularity, the path selection algorithm, executed as step 7 in Figure 16.3, works as follows:

1. Create set *S* of switches containing those switches that are part of at least one input path.
2. Unmark all switches in all input paths.
3. Select and remove the switch with the highest utilization metric from *S*.
4. Mark the selected switch in all paths containing it.
5. If all switches in a path are marked, output that path, otherwise, go to 3.

This approach ensures that switches with the highest utilization are always preferred for placing additional flows, and the least utilized switches remain with no additional flows to increase their chances of being freed from traffic eventually.

The path determined by the aforementioned algorithm is then forwarded to the control module (step 8 in Figure 16.3), which enforces the path after ensuring that all network switches are active (step 9). Although this is usually the case, the selected path may contain network elements that need to be started up first.

In step 10, the route is finally set up by appropriately configuring all switches involved in the path. In the example in Figure 16.3, new traffic forwarding entries are created in switches S_7, S_3, and S_{10}.

The main feature of this approach is that the path selection algorithm looks at the utilization of both the network and the computing domains. By having the load balancer (computing domain) propose multiple potential destination hosts, the set of candidate paths (network domain) is increased. With it, the chances of aggregating traffic are increased and so are the chances of relieving switches from traffic and shutting them down. The joint consideration of the computing and network domains therefore results in better decisions and increased energy savings than could be achieved by looking at the individual domains separately.

16.4 Performance Evaluation

The key effect of the proposed method is that network flows are consolidated on as few network elements as possible while observing all performance constraints. The efficiency of the approach has been evaluated by simulation. We developed an event-based simulator that is designed to track energy consumption of a data center's constituting components. The simulator operates on descriptions of the physical data center model and models of the applications and their traffic. The physical model is described in terms of network and server resources as well as their interconnections. The application models describe behavior in terms of how much load applications put on servers and how much traffic they generate in response to incoming requests. Traffic is modeled with packet granularity. Load is put on the data center by clients outside the data center, which send requests to particular applications.

Multiple instances of applications can be deployed on multiple servers in order to cope with large numbers of requests. Load balancers distribute incoming requests among all running instances of an application according to a defined scheme (we used random distribution in our simulations). Deployments of applications can be scaled dynamically according to the size of the work load they receive. If the currently active application instances are close to their saturation points, an auto scaler component automatically deploys additional application instances. Similarly, if load decreases, unnecessary instances are undeployed from servers and removed from the set of application instances. In our experiments, we assume that a maximum of two applications is deployed per server.

For our simulations we set up a fully populated 24-ary fat tree, which contains 3456 servers and 720 24-port switches. All switches exhibit OpenFlow behavior. In particular, they forward packets according to a flow table that can be influenced by an OpenFlow controller. Connection to the Internet, and therefore to the clients of deployed applications, is provided via routers connected to the core switches of the network.

We modeled three kinds of applications: Web serving (WEB), video streaming (STREAMING), and computing (COMPUTING). The three applications differ in the amount of processing time they need from servers and how much traffic they produce. For each incoming request, the WEB application puts a short peak of load on the server. It then responds with a small burst of data that mimics the traffic produced by serving a medium-sized Web page (64 kB). In the STREAMING application, each request is followed by a 1-minute stream of data bursts (20 chunks of 1 MB each). The COMPUTING application puts a significant amount of load on a server for 5 seconds before it responds with a packet similar in size to the WEB application. We simulated and analyzed each of the applications in isolation, meaning that only that particular application is deployed and used. In addition, a combined setup (HYBRID) was examined in which all three applications are active in parallel and are requested at a ratio of 2:1:1 with the WEB application receiving twice as many requests as the other two.

The number of requests sent to applications follows a normal distribution, aimed at resembling the traffic load distribution of a 24-hour period. In order to reduce simulation time, we map the 24-hour distribution to 10 minutes of simulated time. During that time, up to 10^6 requests are sent to applications (with 28 distinct request numbers evaluated). For the largest simulation, this equates to a maximum of about 5000 requests per second.

Energy consumption of server and network equipment is described by energy models that detail a component's energy consumption in relation to its load. In addition, we assume that components can be switched to a low power sleep mode in which significantly less energy is consumed than when the equipment is on but idle. Moving into and out of sleep mode takes some time and potentially increases energy consumption during the state migration. Although each piece of equipment can have its own specific energy model, all servers and all switches have the same energy models in our simulations. The energy model of the switches is shown in Figure 16.4. The switch's energy consumption is independent of the traffic that crosses the switch, but starting or stopping it requires time and additional power. Energy consumption of servers was modeled as well but is not presented here.

```
begin switch-energy-model
 idle-power 100
 port-power 2
 sleep-power 2
 start-up-power 120
 shut-down-power 120
 start-up-duration-millis 10000
 shut-down-duration-millis 1000
end switch-energy-model
```

Figure 16.4 Switch energy model

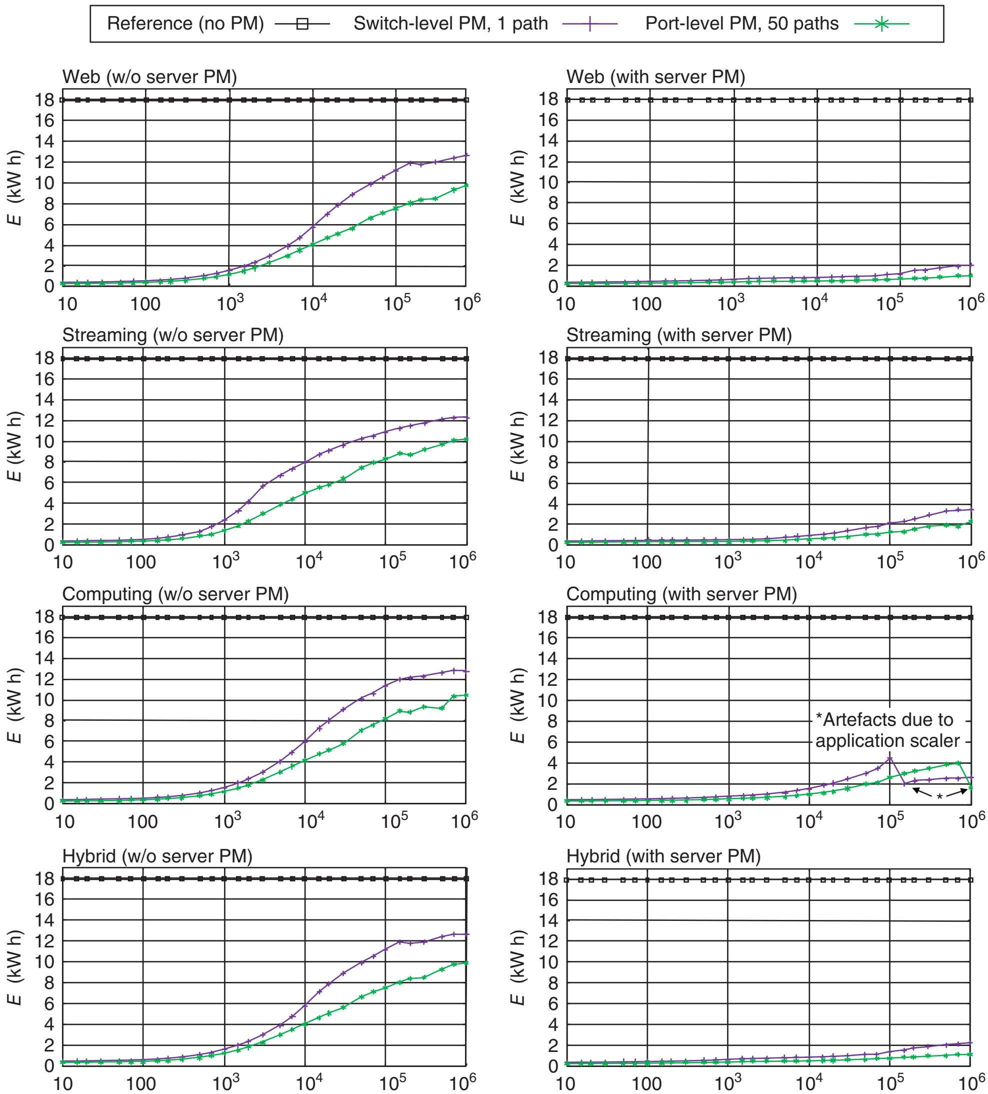

Figure 16.5 Energy consumption E of network equipment in kW h for various scenarios

Figure 16.5 shows the simulation results of the network's energy consumption for each application and the hybrid scenario, aggregated over 10 minutes of simulated time. For each setup, the chart on the left shows the energy consumption when all servers are running continuously and applications are deployed randomly across the data center. Accordingly, the network must ensure connectivity to large parts of the data center. The charts on the right side show the network's energy consumption when server power management is enabled, meaning that servers are shutdown if not needed. Application deployment in this case happens "from left to right" (vs randomly) as servers are gradually activated with increasing load.

Each chart contains three lines. The top line is the energy consumption reference in case all switches are turned on at all times. The middle line corresponds to the case where our energy management strategy is enabled and works on the granularity of switches, turning a switch off when no more flows are allocated to it. The lower line shows the case where energy management is also enabled, but individual ports are turned on and off.

All graphs have in common that applying any kind of network power management significantly reduces energy consumption in accordance with the load on the data center, closely matching an energy-proportional behavior. Simply turning off unused switches (switch-level PM) can yield 25–30% power savings on random application distribution and nearly 90% with ordered application deployment. As explained before, application performance is unaffected. Our proposed power management strategy reduces energy consumption even further when applied on the port level.

When server power management is also enabled, the advantage of a joint energy management approach is evident from a comparison of the graphs on the left and right side of Figure 16.5. It is then possible to achieve even larger energy savings also under heavy load.

16.5 Concluding Remarks

Reducing energy consumption is an important topic for all parts of the IT infrastructure, and data center networks are no exception. Saving significant portions of the energy in either the server or the network domain has been demonstrated by a number of research efforts. With our joint energy management approach we illustrate how the holistic consideration of energy management across domains can help save more energy than by looking at a single domain alone. More research should further explore the possibilities of joint energy management to help make green data centers a reality.

References

[1] J. Koomey, "Growth in Data center electricity use 2005 to 2010." Analytics Press, Oakland, CA, 2011. http://www.analyticspress.com/datacenters.html.

[2] D. J. Brown, C. Reams, "Toward energy-efficient computing," Communications of the ACM, vol. 53, no. 3, pp. 50–58, March 2010.

[3] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, H. Liu, Energy Proportional Datacenter Networks, Proceedings of the 37th Annual International Symposium on Computer Architecture (ISCA'10), pp. 338–347, Saint-Malo, France, June 19-23, 2010.

[4] L. A. Barroso, U. Hölzle, "The case for energy-proportional computing," IEEE Computer, vol. 40, no. 12, pp. 33–37, December 2007.

[5] F. Ahmad, T. N. Vijaykumar, "Joint optimization of idle and cooling power in data centers while maintaining response time," ACM SIGARCH Computer Architecture News, vol. 38, no. 1, pp. 243–256, March 2010.

[6] M. Al-Fares, A. Loukissas, A. Vahdat, "A scalable, commodity data center network architecture," ACM SIGCOMM Computer Communication Review, vol. 38, no. 4, pp. 63–74, October 2008.

[7] P. Reviriego, J. A. Maestro, J. A. Hernández, D. Larrabeiti, "Burst transmission for energy-efficient Ethernet," IEEE Internet Computing, vol. 14, no. 4, pp. 50–57, 2010.

[8] C. F. Lam, H. Liu, B. Koley, X. Zhao, V. Kamalov, V. Gill, "Fiber optic communication technologies: What's needed for datacenter network operations," IEEE Communications Magazine, vol. 48, no. 7, pp. 32–39, 2010.

[9] S. Pelley, D. Meisner, P. Zandevakili, T. Wenisch, J. Underwood, "Power routing: dynamic power provisioning in the data center," ACM SIGARCH Computer Architecture News, vol. 38, no. 1, pp. 231–242, March 2010.

[10] C. Gunaratne, K. Christensen, B. Nordman, S. Suen, "Reducing the energy consumption of Ethernet with adaptive link rate (ALR)," IEEE Transactions on Computers, vol. 57, no. 4, pp. 448–461, 2008.

[11] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, N. McKeown, ElasticTree: Saving Energy in Data Center Networks, Proceedings of the 7th ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI'10), San Jose, California, USA, April 28-30, 2010.
[12] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, D. Wetherall, Reducing Network Energy Consumption via Sleeping and Rate-Adaptation, Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI'08), pp. 323–336, San Francisco, California, USA, April 16-18, 2008.