Energy-Efficient Framework for Networks of Large-Scale Distributed Systems

Anne-Cécile Orgerie and Laurent Lefèvre
ENS de Lyon - INRIA - LIP (UMR CNRS, INRIA, ENS, UCBL)
46 allée d'Italie 69364 Lyon Cedex 07 - France
annececile.orgerie@ens-lyon.fr, laurent.lefevre@inria.fr

***Abstract*—This paper presents HERMES: an energy-efficient data transfer framework for data-center, grid and cloud networks. An architecture and simulations are provided to show that this framework could save more than two-thirds of the energy currently consumed by these networks.**

I. Introduction

In the age of petascale machines and cloud computing, the ever increasing amount of electricity required to power current distributed systems has become a key issue. As these systems grow in number of computing and storage resources, the network has to evolve accordingly.

Advance reservation mechanisms are often used in large-scale distributed systems [1], [2], [3] as a means to guarantee quality of service, help providers meet deadlines and enable users to utilise specific hardware and software resources over well defined timeframes. In networks, advance reservation is used to allocate bandwidth to specific services.

Over the years, networks have become faster, more reliable, and more fault tolerant, but their power consumption has also reached unprecedented figures [4]. Thus far, the main concern when designing network equipments and protocols has been performance, whereas little attention has been paid to energy consumption. With the costly growth in electricity demand, it is important to prioritize energy efficiency in network design.

To this end, we propose a novel and energy-efficient data transfer framework, including scheduling algorithms, which provides adaptive and predictive management of advance bandwidth reservations. This model is called High-level Energy-awaRe Model for bandwidth reservation in End-to-end networkS (HERMES) [5].

Section II discusses background and state of the art. Section III details the architecture of HERMES and its workings. In Section IV, we present a new network simulator: Bookable Network Simulator (BoNeS) and we discuss experimental results. Finally, Section V concludes the paper and proposes directions for future work.

II. Background and Related Work

A. Data Center, Grid and Cloud Networks

Internet and scientific services are often hosted by large computing facilities. Providing these services with high quality and reliability requires robust network infrastructure specially designed for data centers, grids and clouds.

Previous work studied data center networks and traffic in order to develop energy-efficient data center task schedulers [6], [7]. The common scheme is as follows: a centralized scheduler performs the traffic flow optimization inside the data center network [6]. We have opted for a decentralized approach to gain in scalability. The authors of [7] propose traffic-aware virtual machine placement. We have chosen to de-correlate task and traffic scheduling in order to optimize only the traffic scheduling. In fact, these two problems are not equivalent since traffic is not exclusively generated by the computing tasks. For example, in a cloud environment, virtual machine deployment results in data transfers from the server containing the images to the deployed nodes. The network performance of Amazon EC2 was evaluated in [8]. The authors show that virtualization and processor sharing on server hosts lead to unstable network characteristics from the application point of view.

B. Bandwidth Allocation Algorithms

The idea of reserving network resources in advance is not recent [9]. The main issue is the unpredictability of the routing behavior. However, with the emergence of the MPLS (Multi-Protocol Label Switching) [10] standard with traffic engineering and explicit routing features, it becomes possible to disconnect the reservation management from the network layer, thus leading to an easier inter-operability for the ABR management systems. Different data transfer scheduling techniques can be used for advance reservation: online scheduling where requests are processed as soon as they arrive, or periodic batch scheduling where they are scheduled with a certain periodicity [11].

For the moment, none of the proposed solutions consider the network's energy consumption to be a major issue that should influence the design of each algorithm related to the network's management, from scheduling to routing.

C. Green Wired Networking

The energy issue is becoming more and more present in wired networks [12]. The ever-increasing demand in energy can still be greatly reduced. Studies have indeed shown for a few years that network links, and especially edge links, are lightly utilized [12], [13]. This fact has led researchers to propose several approaches to take advantage of link under-utilization in order to save energy.

The first approach, also known as shutdown, consists in switching off (sleeping mode) network equipments when they are not used [14]. This technique raises several problems: connectivity loss, long re-synchronization time, and the fact that constantly switching equipments on and off can be more energy consuming than doing nothing. New mechanisms have been designed to settle these issues, such as: proxy techniques to keep the connectivity [15] and new mechanisms to quickly re-synchronize both ends of a link [16].

When NICs and switches operate at lower data rates, they consume less energy [17]. This observation has led researchers to propose methods to dynamically adjust a link's data rate to the load [18], [19], based on the same principle as Dynamic Voltage Frequency Scaling (DVFS) techniques for CPUs. This kind of technique is commonly termed as slowdown.

By combining shutdown and slowdown techniques according to the scale of the network, we have developed a coordinated model responsible for managing the networks with end-to-end bandwidth reservations in an energy-efficient way.

III. The HERMES Model

A. HERMES Architecture

In the context of grids and clouds data transfers, HERMES is used to manage the entire network. Three traffic characteristics of such systems make them the perfect candidate for HERMES: 1) the traffic stays mainly inside this network: traffic entering and leaving the network represents about 20% [20]; 2) packet arrivals exhibit ON/OFF patterns [21], [22]; 3) the network runs well below capacity most of the time [21], [22].

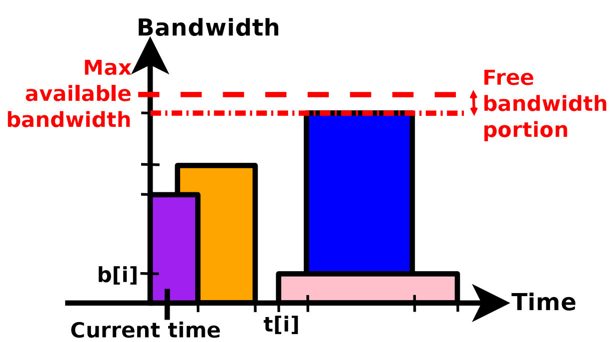

Fig. 1. Agenda example

The first characteristic allows HERMES to have end-to-end bandwidth reservations in order to control the complete network. The second characteristic fosters on/off and traffic aggregation algorithms for energy saving purposes. The third characteristic guarantees that energy savings are feasible by using shutdown and slowdown techniques.

To achieve energy-efficiency, HERMES combines several techniques:

- unused network components are put into sleep mode;
- energy optimization of the reservation scheduling through reservation aggregation;
- minimization of the control messages required by the infrastructure;
- usage of DTN to manage the infrastructure;
- network-usage prediction to avoid frequent on/off cycles.

In HERMES each data transfer between any two nodes of the managed infrastructure (e.g. data center, grid or cloud) should first be submitted to the reservation system. Then, HERMES schedules it, informs the sender about the transfer scheduling, and guarantees that the transfer occurs without congestion. The amount of bandwidth required by the traffic induced by the applications running on the computing nodes is booked when the task is submitted to the infrastructure's task scheduler. At the same time, the data transfers required to launch the application (e.g. virtual machine migration, image deployment) are also submitted to the reservation system of HERMES. Computing tasks can also submit data transfer requests when they are running.

Each network equipment (router, switch, bridge, NIC) has two agendas per port: for both ways (in and out). An *agenda* stores all the future reservations concerning its one-way link. This information is sometimes called the book-ahead interval [23]. Figure 1 presents an example of such an agenda. A *free bandwidth portion* is always kept on each link for management messages and for the ACKs. This portion can be either a fixed amount of bandwidth or a fraction of the link's capacity. Each port maintains its reservation status using a *time-bandwidth list* (TB list) which is formed by $(t[i], b[i])$ tuples, where $t[i]$ is the time and $b[i]$ is the bandwidth available. Each $t[i]$ is called an *event* in the agenda.

B. The reservation process

The reservation process in HERMES is as follows:

1) a user submits a reservation request (specifying at least the data volume and the required deadline) to the network-management system;
2) the advance-reservation environment launches the negotiation phase including admission control, reservation scheduling and optimization policies;
3) the notification is sent to the user when his/her request is accepted or rejected, and when it is scheduled;
4) the reservation starts at the scheduled start time and ends at the scheduled end time, which occurs before the user-submitted deadline.

Each end user is linked to a gateway and know no more about the network. A global view of the reservation process is presented in Figure 2. When a gateway receives a reservation request, the first operation it executes is admission control. The validity of the request is checked. Then, each request requires gathering the agendas of all equipments (ports and routers) along the network paths between the source and the destination.

To perform this collection, the agendas of the possible shortest paths are sent to the gateway of the receiver. The sender gateway sends a particular management message. The first node to receive it, adds its own availability agenda to

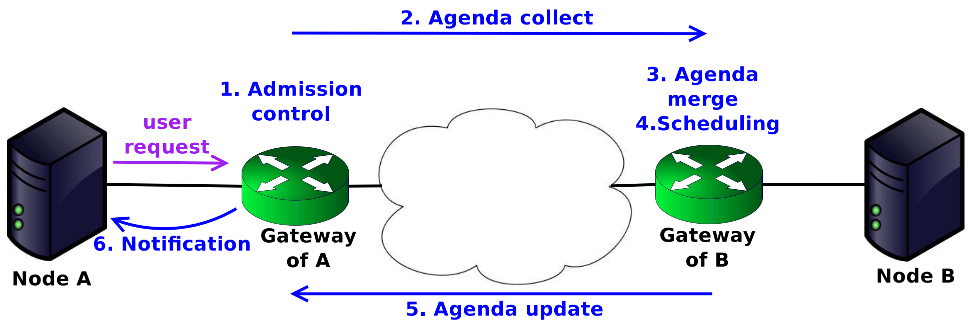

Fig. 2. Reservation process

this message and sends it to the two next nodes which are the nearest to the destination. If the network topology is, for example, a simple tree with no redundant link, only one path is available and thus, the message is sent only to the next node. The agendas of the ports used to transmit these messages are also included in the message. Each node includes the required agenda that it has and passes the message to the next nodes. At the end, the destination gateway re-builds the end-to-end paths.

Thus, the receiver gateway ends up with all the required agendas. It merges the corresponding availability agendas of the nodes to obtain one availability agenda per end-to-end path. Once the agendas are merged, the end-to-end path information is stored in cache in order to avoid doing this computation again. The speed of the merging operation depends linearly on the total number of events in the considered agendas for each end-to-end path. This is fast since the agendas have been truncated to get only the part between the submission time and the deadline.

C. Energy-efficient scheduling algorithm

The end-to-end availability agenda is scanned using the HERMES scheduling algorithm (Algorithm 1) to find the solution consuming the least amount of energy. At each time, the solution tries to use as much bandwidth as it can to reduce the reservation's duration, and thus its cost. We estimate the energy consumption of each possible solution (i.e. place in the agenda), and we compare each solution to pick the least consuming.

D. Resource management

At the end of a transfer between two nodes, if one port is idle for an interval (over a certain threshold), the port is switched off. If all the ports of a router are switched off and the router was supposed to remain idle for a certain period, then the router itself can be switched off. In addition, to avoid unnecessary on/off cycles, prediction algorithms are used to predict the next time a link is used.

The prediction algorithms rely on recent history (past agenda) of the port. They are based on average values of past inactivity period durations and feedbacks which are average values of differences between past predictions and the past corresponding events in the agenda.

Algorithm 1 Scheduling algorithm

```
if the availability agenda of the path is empty then
    Put the reservation in the middle of the remaining period before the
    deadline, if possible. Otherwise, put it now (+ϵ for the request processing
    time).
else
    if there is no event before the deadline then
        Put the reservation in the middle of the remaining period before the
        deadline if possible. Otherwise, put it as soon as possible.
    else
        foreach event in the availability agenda of the path and while it
        occurs before deadline do
            Try to place the reservation after and before the event. Mem-
            orize the possible places (no collision with other reservations
            and end-before deadline).
        if there is no possible place then
            if the reservation can be put before the deadline then
                Put the reservation now (+ϵ for the quest's processing
                time).
            else
                if some events were not possible because of the deadline
                constraint then
                    if the reservation can be put now without respecting
                    the deadline then
                        Propose this solution to the user.
                    else
                        foreach of these remaining events while no so-
                        lution has been found do
                            Try to place the reservation after the event
                            without respecting the deadline. Store the
                            earliest possible place (no collision with
                            other reservations) to propose it to the user.
        else
            foreach possible place do
                Estimate the energy consumption of the transfer using each
                equipment's energy-cost functions.
            if there is one less energy-consuming solution then
                Take that place!
            else
                Take the earliest place among the less energy-consuming
                ones.
```

In addition to these shutdown techniques, HERMES uses Adaptive Link Rate (ALR) during the transfers to dynamically adjust the transmission rate of each port to the used bandwidth. As each transfer is scheduled, the traffic at any time is known. Hence, the complex queue threshold mechanism of ALR, used to change the transmission rate, is not required. The reservation process works only if the necessary ports and routers are switched on when the agenda collection is

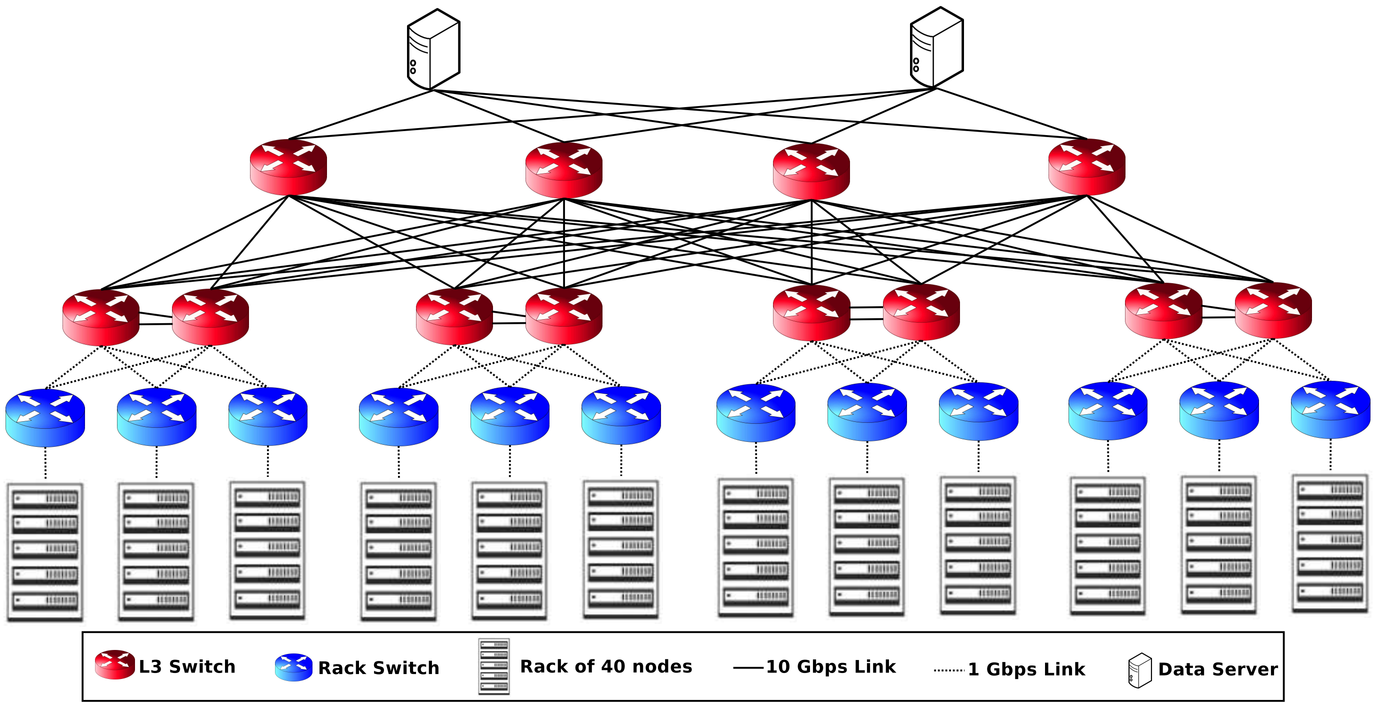

Fig. 3. Typical three-tier architecture

done. Indeed, when they are not used, the network equipments (individual ports or entire routers) are put into sleep mode. To solve this issue, Disruption-Tolerant Networking (DTN) [24] techniques are used. DTN is perfectly adapted to this type of scenario where parts of the network are not always available without any guarantee of end-to-end connectivity at any time.

The idea is to add a kind of Time-To-Live (TTL) in seconds to each end-user request: when the TTL expires, if the request has not reached the receiver gateway and has not returned, then all the sleeping nodes of the path are awaken and the agenda collection is performed. While the TTL is not expired, the agenda-collection message moves forward along the path until meeting a sleeping node. Then, as long as the TTL has not expired, the message waits in the previous node for the sleeping node to wake up. The message is sent to the sleeping node when it wakes up (wake-up detection managed by the DTN protocol) and continues its way. Hence, hop by hop, the agenda-collection message moves towards the receiver gateway.

The gateways are always fully powered on to ensure high availability and reactivity of the overall system. The gateways are able to wake up the nodes to which they are linked. Hence, each sleeping node requires only one awake component (or two if it is connected to a gateway) linked to its manager to be remotely awaken, and not one component per port (i.e. per outgoing link). Compared to a centralized resource management, our approach uses more control messages. But, the number of messages depends on the number of hops in the reservation path, which is quite limited. Moreover, a free bandwidth portion on each link is kept for these messages. As a result, the overhead due to control messages is negligible.

In terms of computational cost, our approach computes several scheduling possibilities for each request. However, we have limited this number to the number of events already put into the agendas and going from the submission time to the deadline, so this number is quite limited too. Moreover, in the worst case, if the reservation is not possible (agendas too busy), our approach has a better complexity than an algorithm that would check all the possible dates. The major advantage compared to a centralized mechanism is the scalability, and thus the reactivity of the whole infrastructure. A more detailed description of HERMES is provided in [5].

IV. Evaluation

A. BoNeS: Bookable Network Simulator

To validate our model, we have designed a network simulator in Python, which is called BoNeS: Bookable Network Simulator (more than 5,000 code lines). It takes as input a network-description file (topology and router and link capacities) and network-traffic characteristics (e.g., statistical distribution of inter-arrival submissions, distribution of the reservation durations, source and destination nodes, distribution of the deadlines and TTLs). It generates the network and an ABR traffic according to the characteristics given as input. It then simulates, with this traffic and topology, different scheduling algorithms and compares them in terms of performance and energy consumption.

Currently, the simulator runs 5 different scheduling schemes on the generated traffic and network: 1) *first*: the reservation is scheduled at the earliest possible place; 2) *first green*: the

Scheduling	First	First green	Last	Last green	Green	No off
Average (Wh)	6 111	6 039	5 684	5 625	5 944	21920
Standard deviation	97	93	76	70	84	371
Accepted volume (Tb)	141.98	141.54	120.24	113.70	141.97	141.98
Cost in Wh per Tb	43.04	42.66	47.27	49.47	41.87	154.39

TABLE I
ENERGY CONSUMPTION IN WH FOR 20% WORKLOAD

Scheduling	First	First green	Last	Last green	Green	No off
Average (Wh)	7 111	6 973	6 300	6 285	6 590	20 463
Standard deviation	362	335	100	106	305	809
Cost in Wh per Tb	42.18	41.37	40.21	41.25	39.09	121.37

TABLE II
COST IN WH PER TB FOR 60% WORKLOAD

reservation is aggregated with the first possible reservation already accepted (before deadline), or scheduled at the earliest possible place; 3) *last*: the reservation is scheduled at the latest possible place (before deadline); 4) *last green*: the reservation is aggregated with the latest possible reservation already accepted (before deadline); 5) *green*: the implementation of our framework: the energy consumption is estimated for each possible allocation, and the least consuming one is chosen.

Our simulator evaluates the energy consumption under these five scheduling schemes combined with our on/off algorithm where resources are switched off when they are not used. The simulator also computes the energy consumption of the *first* scheduling without any on/off algorithm but with ALR (as it could be the case presently). This case is called *no off*. The generated network traffic consists of requests with:

- submission times distributed according to a log-normal distribution;
- data volumes generated with a negative exponential distribution;
- sources and destinations chosen randomly (equiprobability);
- times between submission times and deadlines generated following a Poisson distribution.

The probability distributions of the different traffic characteristics are parameters that can be changed. The distributions presented here are those that we have used in the experiments described next. They have been inspired by the results presented in [25].

The energy consumption of a network equipment (i.e. switch or router) depends on: the type of equipment, the number of ports, the port transmission rates (with ALR), and the employed cabling solutions [26]. Thus, for each router, we have modeled the energy consumption with two values for the chassis power ($P_{chassis}$) depending on whether it is on or off (150 W and 10 W respectively for a 1Gbps router, for instance). We take several values for the port power P_{port}: one for when it is off, one for when it is idle (working at the lower transmission rate), and one for each possible transmission rate. Time to boot and to shut down ports and routers are taken into account.

As we only focus on networking equipment, we do not take into account the energy consumption of servers, we only consider the energy consumed by their Ethernet card.

B. Results

The topology used to evaluate HERMES with BoNeS is described in Figure 3. It represents a typical three-tier fat-tree architecture [6], [27]. Two data servers, which contain for example the images or virtual machines to deploy on the nodes, are directly connected to the core network. This topology comprises 482 servers (including 2 data servers), 24 routers and 552 links.

Two types of traffic are simulated on this network: the transfers among the nodes themselves, this traffic is induced by the user's applications; and the transfers between the data servers and the computing nodes. For each experiment, the simulation has been launched 80 times with requests generated as explained previously. Each simulation represents the behavior of the network during one hour of real time. Table I shows the results obtained with a workload of 20% utilization of the links (i.e. links are used at their full capacity both ways during 20% of the time).

The *last green* scheduling is the least energy consuming. Yet, it accepts 20% less requests in data volume than the *green* scheduling. The *first* scheduling has almost the same percentage of accepted volume, yet it consumes 95 Wh more than the *green* scheduling. This is just for a one-hour simulation. Thus, the *green* scheduling presents the best trade-off between energy savings and request's acceptance rate as shown by the cost in Wh per Tb. Table II shows the results obtained with a workload of 60% utilization of the links. We only present the cost in Wh per Tb since it is the meaningful value to compare the schedulings.

By comparing Tables I and II, one can notice that increasing the link's workload does not affect the performance of the schedulings in terms of cost in Wh per Tb. For a 60% workload, the *green* scheduling is still the best option in terms of both energy consumption and accepted volume. In both cases, energy savings that could be made using HERMES (*green*) are more than two-thirds of the energy consumed in the case of current infrastructures (*no off*): 68% of energy savings with 60% workload and 73% with 20% workload.

V. Conclusion and Future Work

This paper presents HERMES: an High-level Energy-awaRe Model for bandwidth reservation in End-to-end networkS. This framework ensures energy efficiency through energy-aware scheduling with reservation aggregation, and on/off mechanisms for resource management with usage prediction to switch off unused resources.

Using our simulator BoNeS (Bookable Network Simulator), we present an evaluation of HERMES on a typical three-tier network architecture. This evaluation gives really encouraging results: more than two-thirds of the energy used by current data center, grid and cloud networks could be saved using HERMES. These promising results are based on technologies that are not yet fully available on current network equipments; hence, they should encourage manufacturers to design a new generation of network equipments with on/off and slowdown capabilities.

Our future work will focus on exploring more scheduling algorithms, and in particular, off-line algorithms that lead to better optimizations. Such algorithms can be used by discretizing the time in short intervals, and by launching the scheduling at the end of each interval.

References

[1] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, “Resource Leasing and the Art of Suspending Virtual Machines,” in *Conference on High Performance Computing and Communications (HPCC)*, 2009, pp. 59–68.

[2] C. Castillo, G. Rouskas, and K. Harfoush, “Efficient resource management using advance reservations for heterogeneous Grids,” in *International Symposium on Parallel and Distributed Processing (IPDPS)*, 2008, pp. 1–12.

[3] C. Palansuriya, M. Buchli, K. Kavoussanakis, A. Patil, C. Tziouvaras, A. Trew, A. Simpson, and R. Baxter, “End-to-End Bandwidth Allocation and Reservation for Grid applications,” in *Conference on Broadband Communications, Networks and Systems (BROADNETS)*, 2006, pp. 1–9.

[4] M. Baldi and Y. Ofek, “Time For A ”Greener” Internet,” in *GreenComm 09: International Workshop On Green Communications*, 2009.

[5] A.-C. Orgerie, L. Lefèvre, and I. Guérin-Lassous, “Energy-Efficient Bandwidth Reservation for Bulk Data Transfers in Dedicated Wired Networks,” *Journal of Supercomputing*, 2011.

[6] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: Dynamic Flow Scheduling for Data Center Networks,” in *Symposium on Networked Systems Design and Implementation (NSDI)*, 2010.

[7] X. Meng, V. Pappas, and L. Zhang, “Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement,” in *INFOCOM*, 2010, pp. 1–9.

[8] G. Wang and T. Ng, “The Impact of Virtualization on Network Performance of Amazon EC2 Data Center,” in *INFOCOM 2010*, 2010.

[9] W. Reinhardt, “Advance Reservation of Network Resources for Multimedia Applications,” in *International Workshop on Multimedia (IWAKA 1994)*, 1994, pp. 23–33.

[10] E. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol Label Switching Architecture,” RFC 3031, 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3031.txt

[11] Y. Li, S. Ranka, and S. Sahni, “In-advance path reservation for file transfers In e-Science applications,” in *IEEE Symposium on Computers and Communications (ISCC 2009)*, 2009, pp. 176–181.

[12] K. Christensen, C. Gunaratne, B. Nordman, and A. George, “The next frontier for communications networks: power management,” *Computer Communications*, vol. 27, no. 18, pp. 1758–1770, 2004.

[13] A. Odlyzko, “Data Networks are Lightly Utilized, and will Stay that Way,” *Review of Network Economics*, vol. 2, 2003.

[14] M. Gupta and S. Singh, “Greening Of The Internet,” in *SIGCOMM ’03*, 2003, pp. 19–26.

[15] B. Nordman and K. Christensen, “Proxying: The Next Step In Reducing IT Energy Use,” *Computer*, vol. 43, no. 1, pp. 91–93, 2010.

[16] M. Gupta and S. Singh, “Dynamic Ethernet Link Shutdown For Energy Conservation On Ethernet Links,” in *International Conference On Communications (ICC ’07)*, 2007, pp. 6156–6161.

[17] C. Gunaratne, K. Christensen, and B. Nordman, “Managing Energy Consumption Costs In Desktop PCs And LAN Switches With Proxying, Split TCP Connections, And Scaling Of Link Speed,” *International Journal Of Network Management*, vol. 15, no. 5, pp. 297–310, 2005. [Online]. Available: \url{{http://www.icsi.berkeley.edu/pubs/networking/sconn-hotnets07-web.pdf}}

[18] L. Shang, L.-S. Peh, and N. Jha, “Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks,” in *International Symposium on High-Performance Computer Architecture (HPCA03)*, 2003.

[19] M. Bennett, K. Christensen, and B. Nordman, “Improving The Energy Efficiency Of Ethernet: Adaptative Link Rate Proposal,” 2006. [Online]. Available: \url{{http://www.ethernetalliance.org/files/static_page_files/D1481ED9-1D09-3519-AD4429946BB3482E/IMPROVING\%20THE\%20ENERGY.pdf}}

[20] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta, “VL2: a scalable and flexible data center network,” in *ACM SIGCOMM conference on Data communication*, ser. SIGCOMM ’09, 2009, pp. 51–62.

[21] T. Benson, A. Anand, A. Akella, and M. Zhang, “Understanding data center traffic characteristics,” in *ACM workshop on Research on enterprise networking*, ser. WREN ’09, 2009, pp. 65–72.

[22] T. Benson, A. Akella, and D. Maltz, “Network traffic characteristics of data centers in the wild,” in *Conference on Internet measurement*, ser. IMC ’10, 2010, pp. 267–280.

[23] L.-O. Burchard, “Networks with Advance Reservations: Applications, Architecture, and Performance,” *Journal of Network and Systems Management*, vol. 13, no. 4, pp. 429–449, 2005.

[24] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald, “When TCP Breaks: Delay- and Disruption- Tolerant Networking,” *IEEE Internet Computing*, vol. 10, no. 4, pp. 72–78, 2006. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1704758

[25] D. Ersoz, M. Yousif, and C. Das, “Characterizing Network Traffic in a Cluster-based, Multi-tier Data Center,” in *International Conference on Distributed Computing Systems (ICDCS)*, 2007.

[26] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, “A Power Benchmarking Framework For Network Devices,” *Networking 2009*, vol. 5550, pp. 795–808, 2009. [Online]. Available: \url{{http://www.springerlink.com/content/v670rk73668j778v/fulltext.pdf}}

[27] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “ElasticTree: saving energy in data center networks,” in *USENIX conference on Networked systems design and implementation*, ser. NSDI’10, 2010.