Opportunities and Challenges of Optimizing Energy Consumption in Communication Networks using Software Defined Networking

Samar Sabie
Department of Computer Science
University of Toronto
Toronto, Canada
ssabie@cs.toronto.edu

Mahtab Sabet
Department of Computer Science
University of Toronto
Toronto, Canada
mahtab@cs.toronto.edu

***Abstract*— Switches, links, routers, and other network elements draw globally over 350 billion kWh, or 1.8% of the total energy consumption worldwide [1]. This energy consumption will only increase in coming years, as the number of Internet users is projected to grow by 20% per year. In regions such as the Middle East and Africa, this growth rate is even higher. Managing the energy consumption of network elements in communication infrastructure therefore has recently become an important consideration for technical, economic, and environmental purposes.**

Software-Defined Networking (SDN) is a new paradigm in networks that separates the data forwarding plane from the control plane, thus enabling programmability on the global network level, and creating unprecedented opportunities for reducing energy usage. In this paper, we summarize recent work done in energy optimization which builds upon the control and dynamic rerouting afforded by the SDN paradigm. The selected methods are ElasticTree, CARPO, Honeyguide, EQVMP, REsPoNse, and Dynamic Traffic. We then present 12 metrics on which to analyze and compare these methods in order to identify opportunities and common challenges when exploiting software defined networking for energy optimization in communication networks. Based on the comparative analysis, we extrapolate several characteristics for practical energy optimization that collectively mitigate the negative aspects of the work reviewed while accentuating the positive aspects. We conclude by prescribing a set of traffic and network conditions for which the chosen characteristics can be easily and effectively implemented to save at least 25% of the operational energy in SDN-enabled networks.

***Keywords*— energy optimization; software defined networking; traffic engineering; green networking**

I. INTRODUCTION

Human-caused climate change is one of the largest and most complex problem humanity will have to face in upcoming years. While the solution is simple - reduce greenhouse gas emissions such that the amount of CO2 in the atmosphere is below 350 parts per million - implementation is not so easy [2]. Many researchers have focused their efforts on ways to reduce greenhouse gas emissions in existing industries, yet emissions worldwide continue to increase as conflicting interests impede progress in this area.

The environmental impact of Information and Communication Technology (ICT) is no longer negligible, especially as the sector continues to expand exponentially. For example, when it comes to networks, expectations of speeds and reliability are increasing: video streaming services have made 1080p the norm; we expect our browser and search engines to auto-complete our search queries; and we expect our data to be accessible instantaneously. As information and communication technology devices become more commonplace, reducing the power consumption in networks and data centers becomes an area in which small improvements in energy efficiency can make a large difference. While increasing the efficiency of energy consumption has been traditionally attained through continuous hardware improvements (better circuits, reduced friction, simpler components), the advent of software defined networking affords further room for improvements through traffic engineering, virtualization, and rule placement as we discuss in the next sections.

II. SOFTWARE DEFINED NETWORKING

In typical data communication networks, most components such as routers and switches are built as closed systems that are proprietary, static, and vendor-specific. Moreover, control and data forwarding are tightly coupled, as the flow of packets is orchestrated on-board each network element. While such architecture has so far afforded simple and rapid deployment frameworks to meet the explosive demands for communication infrastructure, it imposes several limitations. For example, all forwarding elements must be sophisticated and expensive in order to handle the required routing algorithms. Furthermore, new protocols and functionalities are cumbersome to propagate, and must be implemented directly into each piece of hardware [3]. Finally, the lack of common interface and global control over the network renders the execution of tasks such as dynamically turning off unused elements to reduce energy consumption hard if not impossible.

Software defined networking remedies the aforementioned

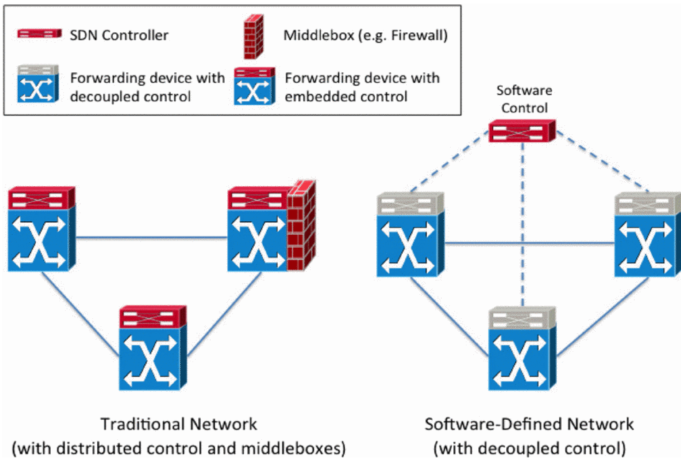

Fig. 1. SDN architecture. Dashed lines designate links between the controller and forwarding hardware (data plane). Solid lines designate links between the data planes [3]

challenges through its fundamentally different architecture. As visualized in Figure 1, this emerging paradigm separates the control and data planes by introducing a centralized logic entity (controller) which configures the forwarding tables, among other actions, and relays them to programmable network devices (switches/simple forwarding hardware).

The flexibility, efficiency, abstraction, and global control brought by SDN to communication networks means that administrators can configure, manage, and optimize network resources quickly and dynamically through third party applications such as firewalling and load balancing. But beyond trivial functionalities, the programmability of SDN can also foster reductions in energy consumption through innovative software-based methods that, when coupled with hardware improvements such as low-power CPUs and more efficient power supplies, leads to significant savings as shown in the next sections. Finally, SDN is growing in popularity and is expected to be widely deployed in campus networks, data centers, and mobile networks [1]. Therefore, considering the potentials of energy optimization in SDN is both timely and worthwhile in order to meet growing networking needs in an environmentally sustainable manner.

III. ENERGY CONSUMPTION TRENDS IN COMMUNICATION NETWORKS

The energy demands of the ICT sector cannot be only articulated by the electricity it consumes, but also by the energy needed for its manufacture, use, and disposal [4]. As such, when considering the energy efficiency of networks, both embodied and operational energy must be taken into account. Embodied energy is defined as the energy required to manufacture an item, which includes resource extraction, the manufacturing process, and transportation. Operational energy is the energy required during a product's use [5, 6]. For the purposes of this paper, we consider both energy aspects but not a product's end-of-life process, which involves e-waste collection.

In 2008, ICT contributed to 8% of the global energy consumption and is projected to be 14% in 2020. Network devices are responsible for 15% of that. More specifically, the operational energy consumption for data center and network equipment is about 54 GW worldwide according to a 2008 yearly average. This accounts for approximately 2.57% of the global electricity consumption [5]. For small data centers composed of about 100-500 servers, the power input is about 50kW [7]. For large data centers of 5,000+ servers, the power input can reach 50 MW or more [8]. Such rates might seem insignificant. But their cumulative effect is colossal because the number of data centers being built around the world is projected to reach 8.6 million in 2017 according to the International Data Corporation (IDC) recent trends report [9].

The energy intensity of the internet, or the "energy consumed to transmit a given volume of data", is hard to accurately measure. For Home and Access Networks, estimates vary between 0.0064 kWh/GB up to 136 kWh/GB, depending on the scope and boundary considered, and the year of the study [10]. For Edge and Core Networks, Schien et al [11] propose a model that estimates an energy intensity of 0.052 kWh/GB. Either way, the energy consumption will only increase in coming years, as the number of Internet users is projected to grow by 20% per year worldwide [12]. Figure 2 depicts a similar trend in terms of traffic, even in areas with the least internet usage today such as Africa. In fact, Cisco's forecast projects that IP traffic is growing the fastest in the Middle East and Africa, with a Compound Annual Growth Rate (CAGR) of 44 percent between 2014 and 2019. At such rate, IP traffic in the Middle East and Africa will reach 9.4 exabytes per month by 2019, up from 1.5 exabytes per month in 2014 [13]. This volume translates to 64 to over 300 GW of consumed power, thus surpassing in those two regions alone the global network energy consumption in 2008 by several folds.

In addition to the exponential growth, a major problem with the consumption rates reported above is that a significant portion of the used energy is unjustified waste. This is due to networks running constantly at maximum capacity to accommodate for rarely-occurring worst-case workloads [14]. To elaborate, communication networks transport and deliver data reliably in part due to the redundancy and bandwidth overprovisioning of their components. This leads to power consumption independent of the workload, as network devices remain on and draw nearly constant power even in absence of traffic, resulting in the dis-proportionate demand to consumption graphs of Figure 3. Even when they are not forwarding any packets, they still draw 90% of their maximum power consumption [15]. Moreover, studies show that networks, particularly data centers, run well below capacity most of the time [16]. Accordingly, over the past decade, many network energy optimization strategies have surfaced focusing specifically on inducing energy consumption proportionality into networking devices. Six of them will be discussed in the subsequent sections.

IV. ENERGY OPTIMIZATION METHODS FOR SDN: OVERVIEW

Reducing energy consumption in networks is a function of minimizing the overall energy consumption and introducing

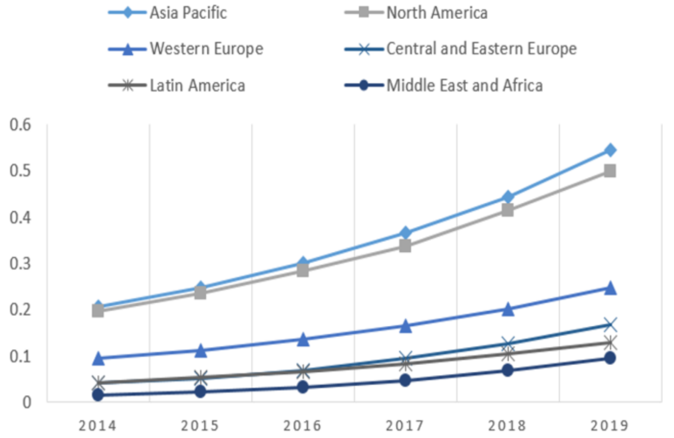

Fig. 2. Wordwide IP traffic growth 2014-17 in exabytes based on data from Cisco Visual Networking Index

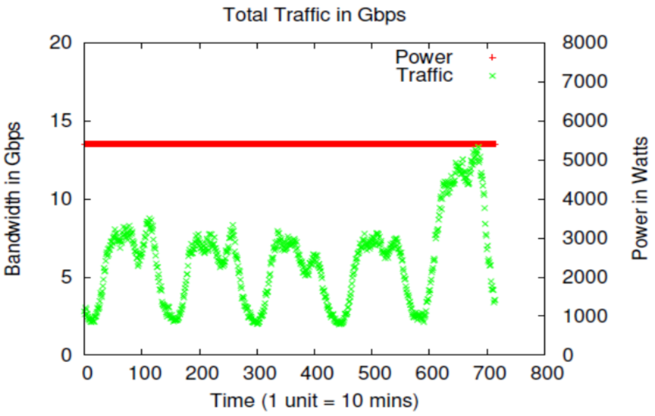

Fig. 3. Typical netwrok traffic fluctionations (based on traces from an E-commerce website) and constant energy consumption [16]

energy proportionality. This process can be carried out through many strategies (in software and hardware) implemented over all network components such as:

- Switches (reducing number, size, power usage, state, controlling clock rate, …)
- Servers (optimizing virtual machine placement, minimizing power usage in physical machines, …)
- Traffic Paths (shortest paths, smallest number of paths, least energy-expensive paths...)
- Rule Placement (minimizing re-routing rates, comparing flow tables, …)
- TCAM architecture (minimizing memory needs, reducing information exchange, ...)
- Links (rate, state, capacity, ...)
- Auxiliary components (fans, cooling pipes, building envelopes, ...)

The large body of scholarly literature on optimizing energy consumption in communication networks employs one or more of the aforementioned techniques for reducing energy consumption in networks and data centers. We review six recent strategies that are software-based, i.e. they do not work on optimizing the actual server, switch, or TCAM hardware. While not all of them are designed specifically for SDN, they all require one form or another of a central control logic and a global view of the network, which makes them ideal for deployment in software defined networks. Our selection criteria is based on both approach and popularity. Approach-wise, we chose methods that work on switches (reducing their number/maximizing sleep time), servers (optimizing virtual machine placement), traffic paths (finding least energy-expensive routes), and rule placement (re-routing rates). We chose a combination of popular and unpopular work in terms of impact and the number of times it was cited in order to cover a wider range of known and unknown gems as follows: Elastic tree (cited on Google Scholar 509 times), CARPO (65), REsPoNse (53), EQVMP (7), Honeyguide (13), and Dynamic Traffic (1).

One way of reducing energy consumption is to switch off network elements during low traffic periods. ElasticTree and Honeyguide utilize this approach. ElasticTree [16] is a method for saving energy in data center networks. Because network elements consume energy at a constant rate regardless of network traffic, energy is being unnecessarily wasted on powering idle and underutilized switches. ElasticTree works by monitoring the network traffic in data centers, then based on the performance and fault tolerance requirements, decides which network elements must stay on, and which can be shut off. The authors use different algorithms for determining which network elements can be shut off, including a formal model, greedy bin-packer, topology-aware heuristic, and prediction. The authors suggest that the network energy consumption could be reduced by 25-40% on average using this method. One constraint of ElasticTree is that it would not work for a highly-utilized never-idle data center because the minimum-power network subset will be equivalent to the entire set of elements.

CARPO, or Correlation-aware power optimization [14] is similar to ElasticTree in that it dynamically finds a subset of network elements that can be shut off without adverse impact to performance requirements. It does this by consolidating traffic flows into a subset of network elements based on whether or not flows are positively correlated (i.e. peak at the same time). This leads to better performance and increased reliability since different flows usually do not surge at exactly the same time, so the current subset capacity is more likely to be sufficient until the next update.

Shirayanagi et al [17] propose Honeyguide, another method for optimizing energy consumption in data center networks by turning off unused network switches. This is done through a combination of virtual machine (VM) migration, traffic consolidation, and special links added between servers and upper tier components to bypass edge switches that have been turned off. One important characteristic of Honeyguide is that fault tolerance of existing tree-based topologies is maintained because of the bypass links. There is no need to replace existing network topologies; Honeyguide can be deployed by creating additional links. Simulations suggest energy savings of up to 7.8%.

In their paper on EQVMP (Energy-efficient and QoS-aware Virtual Machine Placement), Wang et al [18] describe the process by which they "overcome the problem of unbalanced traffic load in switching on and off VMs for the purpose of energy saving." This method takes as input VM resource demands, VM traffic, and a topology matrix, and utilizes hop reduction, energy saving, and load balancing to achieve EQVMP. Server resources are not always used to their full capacity. Instead of placing 4 VMs on 4 different servers, which would result in energy consumption of about 800 watts, they suggest placing all 4 VMs on the same server, and load balancing, resulting in energy consumption of only 300 watts.

The goal of REsPoNse [19] is to precompute energy-critical paths in an arbitrary topology by analyzing its past traffic matrices. Then large parts of the network can enter a low-power state by identifying a few energy-critical paths off-line, installing them into network elements, and using an online element to redirect the traffic. REsPoNse precomputes three sets of paths. The "always-on" path is the set of network elements that are expected to be on all of the time. The "on-demand" path are those elements that can be switched off to save energy, and switched back on for additional network capacity. The "fail-over" path can carry traffic in case of failure in the other two scenarios. The limitation of this method is that it only works for predictable traffic with known historical stats.

Lastly, Markiewicz et al [1] discuss energy optimization considering dynamic traffic. Similar to the methods discussed above, the goal is to switch on the minimum number of network elements to support network traffic, but through using fast non-iterative algorithms that work for arbitrary topologies, not just for data centers, and prevent degradation of the network performance. Their results showed that during low-use times (i.e.at night), their method can save up to 45% of energy consumption.

V. ENERGY OPTIMIZATION METHODS FOR SDN: ANALYSIS

In addition to the six works summarized above, we found over thirty published research studies just within the last five years proposing strategies for reducing network energy consumption. Many share the same conceptual framework of making the network energy consumption proportional to its utilization, be it on the link, switch, and/or server levels. The problem is that no single strategy yet has proved sufficient to effectively achieve the desirable energy optimization. Indeed, upon a thorough examination of the six selected proposals, we found that despite the diverging approaches employed (elastic topologies, traffic rerouting, and virtualization), they all seem to have common shortcomings. For example, they cause brief bottlenecks when network traffic volume increases, or induce unusually high packet drop rates due to the use of complex energy-aware algorithms to update the flow tables. Yet, we found that amidst the similarities and differences in their strategy and performance lies a set of opportunities (achieving considerable energy savings on predictable traffic patterns for example) that could be instigated and built upon to make the reduction in network energy consumption more attainable. To elucidate the review process, we propose and define 12 metrics against which to analyze, compare, and contrast the chosen energy optimization strategies. The goal of the metric-based comparison is not to solely critique the methods, but rather to identify "commonalities" and "best practices" in order to derive a better model. As such, this critical review brings out the strength and weaknesses of every strategy and where they could be combined to complement each other. Moreover, it is the basis for developing the final set of recommendations that integrate multiple strategies to achieve maximum energy optimization while minimizing the undesirable side effects.

The following lists the 12 metrics we propose:

1) *Strategy*
2) *Algorithms*
3) *Target Topology*
4) *Traffic Type*
5) *Update Intervals*
6) *Ease of Implementation*
7) *Scalability*
8) *Embodied Energy Considerations*
9) *Redundancy Considerations*
10) *QoS Considerations*
11) *Environmental Performance*
12) *Impact on Research Community*

Strategy refers to the general approach the chosen methods employ to achieve energy optimization. This includes dynamically finding a minimal subset of ON-components (links, switches, and VMs), finding an efficient grouping of VMs, precomputing energy-critical subsets and paths, and/or dividing traffic into equal flows to increase the utilization of ON-components. The algorithm metric means the actual algorithm(s) the strategy uses such as optimal, greedy, or heuristic, whether they are original or modified from common algorithms, and their runtime. A strategy that saves energy only through formal algorithms that cannot be efficiently computed on a larger scale loses merit when it comes to feasible implementation. Target topology refers to the generalization of the strategy, and which topology type the method can function on such as datacenter, campus, and/or backbone networks. Because we consider real data center networks, we assume that it is a full mesh topology. Traffic type examines the method's applicability to continuous/discrete, as well as high, medium, and low traffic. Update intervals refers to the nature of computing and routing table update, be it static or dynamic, short or long, and how the strategy responds to traffic changes. Ease of implementation, scalability, embodied energy considerations (i.e. reducing the number of physical units), redundancy considerations, QoS considerations, and environmental performance (how the strategies live up to their goals) are other metrics we will consider. In the final metric, we investigate the number of papers referencing the chosen method, other strategies based on it, and subsequent work by the authors to qualify its influence on the community and hence overarching impact. The full comparison table is attached in Appendix A.

VI. ANALYSIS RESULTS

A. *Comparison Summary*

In this section, we present a summary of findings and analysis of the methods' comparison against the metrics defined in the previous section. Please refer to Appendix A for the full table. In terms of strategy, all methods are based on learning traffic behavior and changing the network topology according to it, such that a minimal number of components (links, switches, and/or servers) is on. The change in topology is propagated by updating the routing tables. All methods do it dynamically at varying rates, from ElasticTree that re-computes every 5 minutes to REsPoNse which pre-computes the routing tables only once every 7 to 10 days. Honeyguide and EQVMP control VM placement in addition to changing the topology. The majority of algorithms used are modified versions of popular Bin-Packing, First-Fit, and Load Balancing. All methods are implementable on the scale of data centers (full mesh) topology with any traffic type that does not fluctuate very frequently. Therefore, they are all scalable and implementable with minimal to no hardware changes in the network. All methods only discuss the operational energy consumed during the use stage of a product, and none consider embodied energy, which is the energy required to manufacture an item, including resource extraction, the manufacturing process, and transportation. In fact, Honeyguide and Elastictree require/recommend the use of extra components to increase reliability and QoS.

B. *Energy Performance Evaluation*

Based on a comprehensive examination of the environmental performance analysis of the methods, we found that the numbers presented in Appendix 1 row 12 (best environmental performance claimed) are only true for rare or specific cases such as for traffic at night, with no redundancy considerations, or when running unsalable formal algorithms. If considered for average traffic, during the middle of the day for example, the numbers are actually much lower.

Honeyguide quotes 7% energy savings after its tests on a synthetic workload derived from surveys of real data center traces with resource utilization ranging between 5% and 60% (peak). Yet, the 7% only holds in a very specific case: at k=12, i.e. when the datacenter is a fat tree with 432 physical machines, 72 racks, 648 service VMs, and 180 replica VMs (running on racks separate from their respective service VM). Any increase or decrease in the number of VMs, ratio of service to replica VMs, or the value of k, and energy savings drop to 1-3% as evident in the performance charts the authors provide [17]. The test evaluations of Elastictree were performed using a wide range of datacenter traffic types and volumes including uniform, random, discrete, continues, far, and near. The reported savings are 20-64%, attained using the formal algorithm which does not scale and is computationally expensive. As such, using the more feasible greedy algorithm or scalable heuristic will lead to lower savings as the authors admit [16]. The actual range of savings we estimate therefore, based on performance examples of Elastrictree's greedy algorithm, is 25-40% if not lower. While these energy savings are still significant, they are expensive: Elasictree tends to introduce bottlenecks when traffic surges, rapidly or not, beyond the current minimal set of resources. Over-allocating resources as a safety margin for maintaining QoS ultimately lowers energy savings, making the above range only applicable for stable/predictable traffic (which does not require a significant safety margin).

CARPO was shown to save 46% of network energy for a DCN with Wikipedia traces without major increases in delay or packet drop rate. We cannot judge how CARPO performs in general because the experiments presented in the paper [14] were only performed on one type of traffic, and were shown as plots of macro time (in days) to energy saving rather than traffic volume or time (minutes or hours) to energy savings. Since CARPO takes traffic correlation into account to increase energy savings and throughput, it is an improved version of Elasictree. Therefore, we expect average savings closer to 41-46% claimed. REsPoNse uses historical traffic data to pre-compute routing tables for the next week or two. Its power consumption in large ISP networks as well as small FatTree data center networks is shown to be 60-80% of the original consumption. Because of its reliable approach and the wide range of evaluation experiments, we estimate REsPoNse's average savings as a solid 30%. The evaluation experiments of EQVMP focus mostly on how its throughput outperforms other VM placement algorithms such as Max-Min Multidimensional Stochastic Bin Packing, Traffic-aware VM Placement, and Random VM Placement. One small chart shows that EQVMP is one of the superior VM placers in terms of energy performance but no explicit percentages are given [18]. Finally, Dynamic Traffic saves up to 45% of energy consumption but only for nighttime traffic in a campus network or a random mesh network of 40 nodes of degree 4 [1]. For an average 600 Gbit/s daytime traffic load, savings drop to 20% in the campus network with the shortest path first (SPF) dynamic algorithm and 40% in the mesh network with the highest demand first (HDF) dynamic algorithm. It should be noted that beyond 600 Gbit/s traffic, reliability degrades significantly and the above algorithms cannot route all traffic. These savings do not account for redundancy. Incorporating it will expectedly lower energy savings.

Row 13 in Appendix A lists our aforementioned estimated averages. The disparity between the energy savings claimed and estimated averages is a reflection of the common challenges of SDN-based energy consumption optimization. These challenges are identified next. Yet, the fact that Elastictree, CARPO, REsPoNse and can still save up to 40% in the average case (when compared to conventional networks with the same size and traffic) prove the optimization opportunities SDN affords. It should be noted that the performance averaging above is based on calculations, educated guesses, and observations extrapolated from the various assessments the authors' of each method present. As such, there is room for error. Qualitative experiments must be done with larger sets of traffic to better estimate and understand the real energy savings.

C. *Metric Analysis and Findings*

All six methods aim to maximize the number of physical network components that can be shut off. Our average environmental performance analysis affords both predictable and unpredicted observations. Expectedly, there is in all cases a trade-off between energy savings, performance, and reliability. Any method that takes QoS and redundancy into account such as Honeyguide does not achieve significant savings. In fact, energy savings in highly utilized and never-idle data center network cannot be attained through software. Except for REsPoNse which uses observed traffic trends to pre-compute and install routing tables for several days in advance, all methods dynamically compute the minimal set of components at high frequencies. The average computation time across methods is 5 to 10 minutes, and grows exponentially beyond a 1,000 components [18, 19]. During that time, the network runs with inadequate causing congestion due to insufficient resources. Other common challenges include:

- Waiting for switches to boot, links to come online, and new routes to be pushed to flow tables can take minutes even in small networks
- The traffic matrix must be known and is assumed to be a constant until the next computation
- No significant energy savings are achieved for heavy/far/inter-pod/changing traffic
- Some algorithms at higher utilizations (>70%) or random traffic even fail to find a complete satisfying assignment of flows to links
- The effect of frequently putting network devices into sleep and waking them up can negatively impact the component's lifespan, necessitating more frequent changes, which may or may not lead to higher consumption of embodied energy that offsets the attained savings in operational energy

D. Recommendations for Practical Implementation of Energy Optimization

Saving energy often comes with trade-offs with reliability, scalability, and performance. But most of the methods reviewed above work well for specific traffic and networks conditions. By combining some of their strategies (elastic topologies, pre-computed energy paths, and machine virtualization), savings of 25-40% can be practically implemented in any SDN mesh network by following the recommendations listed below.

Traffic:

- Target periods of low traffic (night, holidays, etc.)
- Target traffic that takes sine wave shape or is very predictable
- Avoid consolidating traffic flows that are positively correlated (e.g., peak at the same time) based on 90-percentile demands instead of peak demands

Virtualization:

- Design VM servers in a way that pairs VMs with high mutual traffic to the same host with low cost connection and in the same group together
- Use multiple virtual switches on a single physical switch

Note that turning off more switches and physical machines not only saves energy there, but also less heat will be generated in the server room so less cooling is needed. As long as traffic is predictable and low, redundancy and QoS can be accommodated with little impact on energy optimization as shown in the evaluation of Elastictree [16].

VII. CONCLUSION

The various endeavors made by governments, industry giants, and research communities towards optimizing network energy are motivated by many different reasons. One is the substantial cost of wasted energy, estimated at $0.3 billion/year in the US in 2006. Another reason is the constant expansion demands in the networking infrastructure as noted at the beginning of the paper, necessitating increased energy efficiency to make the expansion feasible and practical. A third reason is the need to cut back wasted energy consumption (2.5 TWh in the US, 2006) in order to reduce greenhouse emissions for the sake of preserving the planet for future generations [16,14]. Regardless of the motivation, energy consumption of ICT in general and networks in particular is already significant, and poses many environmental, performance, and economic threats.

Yet reducing energy consumption in networks is a complex problem, and the methods we reviewed have put forward several solutions such as shutting off network elements during low traffic periods, introducing elasticity in network topologies, and using strategic VM placement. But saving energy often comes with trade-offs in terms of reliability, scalability, and performance. Based on our observations from the metrics comparison and analysis, as well as energy savings calculations, we can combine and exploit the advantages of the methods we analyzed, while mitigating the challenges. Namely, in any mesh network that supports SDN, an optimization model can be easily implemented to save energy if the network experiences periods of low traffic (night, holidays, etc.) and/or routes traffic that takes sine wave shape or is very predictable. Avoiding the consolidation of traffic flows that are positively correlated (e.g., peak at the same time), grouping virtual machines with high mutual traffic on the same hosts, and running multiple virtual switches on a single physical switch lead to further savings with little impact on QoS. Of course, the continuous monitoring of traffic must be incorporated in all networks in order to understand traffic tendencies and establish robust predicative routing rules. Deployment of switches, links, and routers that turn on and off faster is also imperative.

Embodied energy is an important consideration for energy optimization methods. None of the methods we analyzed took this into account, as they only discuss operational energy. Given that electronic devices have a high ratio of embodied energy to operational energy (80%:20%) [20], shortening the lifespan of network components due to frequently turning them on and off may lead to higher consumption of embodied energy that offsets the attained savings in operational energy. Therefore, for future work, we plan to investigate the correlation between lifespan and embodied energy, and if there is a minimum threshold of operational energy optimization needed to offset the additional embodied energy created.

ACKNOWLEDGMENT

The authors would like to thank Mohsen Norouzi, Computer Science MASc candidate at UofT, for his generous help and valuable contributions to this research, particularly the technical networks-related details and metrics table. They are also grateful to Dr. Yashar Ganjali and Dr. Steve Easterbrook for their suggestions and feedback to improve the quality of the paper.

REFERENCES

[1] Markiewicz, Adam, Phuong Nga Tran, and Andreas Timm-Giel. “Energy consumption optimization for software defined networks considering dynamic traffic.” Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on. IEEE, 2014.

[2] Rockstrom, Johan, et al. “A safe operating space for humanity.” Nature 461.7263 (2009): 472-475.

[3] Nunes, Bruno Astuto A, Marc Mendonca, Nguyen Xuan-Nam, Katia Obraczka, and Turletti, Thierry. “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks.” IEEE Communications Surveys & Tutorials, Volume 16, Issue 3, pp. 1617 – 1634, 2014.

[4] Ropke, Inge and Toke Haunstrup Christensen. “Energy impacts of ICT - Insights from an everyday life perspective.” In Telematics and Informatics, Volume 29, Issue 4, 2011/2012, p. 348.

[5] Pickavet, Mario, et al. “Worldwide energy needs for ICT: The rise of power-aware networking.” Advanced Networks and Telecommunication Systems, 2008. ANTS’08. 2nd International Symposium on. IEEE, 2008.

[6] Teehan, Paul, and Milind Kandlikar. “Comparing embodied greenhouse gas emissions of modern computing and electronics products.” Environmental science & technology 47.9 (2013): 3997-4003.

[7] Schomaker, Gunnar, Stefan Janacek, and Daniel Schlitt. “The energy demand of data centers.” ICT Innovations for Sustainability. Springer International Publishing, 2015. 113-124.

[8] Hintemann, Ralph. “The Impact of the Changing Structure of Data Centers on Total Electricity Demand.” ICT Innovations for Sustainability. Springer International Publishing, 2015. 125-136.

[9] Data Center Knowledge Report: http://www.datacenterknowledge.com/archives/2014/11/11/idc-amount-of-worlds-data-centers-to-start-declining-in-2017/

[10] Coroama, Vlad C., et al. “The energy intensity of the Internet: home and access networks.” ICT Innovations for Sustainability. Springer International Publishing, 2015. 137-155.

[11] Schien, Daniel, et al. “The energy intensity of the Internet: edge and core networks.” ICT Innovations for Sustainability. Springer International Publishing, 2015. 157-170.

[12] Vereecken, Willem, et al. “Power consumption in telecommunication networks: overview and reduction strategies.” Communications Magazine, IEEE 49.6 (2011): 62-69.

[13] Cisco Visual Networking Index

http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html

[14] Wang, Xiaodong, et al. “Carpo: Correlation-aware power optimization in data center networks.” INFOCOM, 2012 Proceedings IEEE. IEEE, 2012.

[15] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power Awareness in Network Design and Routing. In INFOCOM, pp. 1130-1138, 2008.

[16] Heller, Brandon, et al. “ElasticTree: Saving Energy in Data Center Networks.” NSDI. Vol. 10. 2010.

[17] Shirayanagi, Hiroki, Hiroshi Yamada, and Kono Kenji. “Honeyguide: A vm migration-aware network topology for saving energy consumption in data center networks.” IEICE TRANSACTIONS on Information and Systems 96.9 (2013): 2055-2064.

[18] Wang, Shao-Heng, et al. “EQVMP: Energy-efficient and QoS-aware virtual machine placement for software defined datacenter networks.” Information Networking (ICOIN), 2014 International Conference on. IEEE, 2014.

[19] Vasic, Nedeljko, et al. “Identifying and using energy-critical paths.” Proceedings of the Seventh Conference on emerging Networking EXperiments and Technologies. ACM, 2011.

[20] Ethernet Power Consumption for Cisco ® Switches:

http://www.atrac.com/documents/Cisco/Enterprise/brochures/Cisco_BROCHURE_Ethernet%20Power%20Consumption%20for%20Switches_2.14.12.pdf

[21] Nedevschi, Sergiu, et al. “Reducing Network Energy Consumption via Sleeping and Rate-Adaptation.” NSDI. Vol. 8. 2008.

[22] Hischier, Roland, et al. “Grey energy and environmental impacts of ICT hardware.” ICT Innovations for Sustainability. Springer International Publishing, 2015. 171-189.

[23] http://www.rensmart.com/Information/Library/101006-guidelines-ghg-conversion-factors.pdf

[24] http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-3750-series-switches/productdatasheet0900aecd80371991.html

[25] https://supportforums.cisco.com/discussion/11041096/best-practice-replacement-plan-routers-and-switches

[26] Schroeder, Bianca, and Garth A. Gibson. “Disk failures in the real world: What does an MTTF of 1, 000, 000 hours mean to you?.” FAST. Vol.7. 2007.

[27] Khan, Samee Ullah, and Albert Y. Zomaya, eds. Handbook on Data Centers. Springer, 2015. Page 127.

[28] http://www.infoworld.com/article/2640563/green-it/powering-down-servers-is-a-calculated-risk.html

[29] Ikonen, Mika. “Power cycling lifetime estimation of IGBT power modules based on chip temperature modeling.” Acta Universitatis Lappeenrantaensis (2012).

APPENDIX A: METRIC-BASED COMPARISON TABLE FOR THE SIX METHODS

	Method					
Metric	***Honeyguide***	***ElasticTree***	***CARPO***	***REsPoNse***	***EQVMP***	***Dynamic Traffic***
Strategy	Save energy by maximizing the number of off (inactive) physical servers and edge switches. This is attained by grouping as many VMs on the same physical servers and creating bypass links between upper-tier switches and physical machines in order to turn off unnecessary edge switches while maintaining accessibility to servers through multiple paths	Maximize the number of off (inactive) switches and links by dynamically finding a minimum-power network subset across a range of traffic patterns that meet the current bandwidth demands	Maximize and shut down unused network devices by dynamically consolidating traffic onto a small set of links and switches based on correlation analysis among flows (e.g. different flows usually do not peak at exactly the same time)	Enable large parts of the network to enter a low-power state by identifying a few energy-critical paths off-line, installing them into network elements, and using an online element to redirect the traffic. This is an alternative to computing the minimum network subset dynamically, which is computationally hard, does not scale, and forces the network to operate with diminished performance during the precomputation periods	Lower energy consumption while overcoming unbalanced loads due to switching on and off VMs for the purpose of energy saving through three consecutive step: hop reduction, energy saving and load balancing	Switch on a minimum amount of necessary switches/routers and links to carry traffic using a fast non-iterative algorithm that works for arbitrary topologies and prevents degradation of the network performance
Algorithms	Modified First-Fit	Formal Method, Greedy Bin-Packing, Topology-aware Heuristic, and Predictive	Greedy Bin-Packing	The REsPoNse-Heuristic, REsPoNseospf, and REsPoNseTE	Cluster-&-Cut inspired algorithm, Best Fit Decreasing (BFD), Max-Min Multidimensional Stochastic Bin Packing (M3SBP), Load Balancer	Strategic Greedy Heuristic, Shortest Path First (SPF), Longest Shortest Paths First (LPF), Smallest Demands First (SDF), & Highest Demands First (HDF)
Target Topology	Fat tree topology in data centers	Fat tree topology in data centers	Fat tree topology in data centers and other network topologies	Data Centers and ISPs with mesh network topology	Topology-independent	Data centers (mesh network)
Traffic Type	Continuous, high, and low	Discrete, continuous, random, or periodic as long as it is perfectly divisible and traffic rate of each data flow is approximately a constant	DCN traffic with quantifiable and stable correlation relationships (based on historical records or statistical analysis for example)	Any traffic that does not experience a significant and unanticipated deviation from its long term trends averaged over multiple days (or months)	Any traffic type with a known load matrix between VMs	Any traffic type
Update Intervals	Not specified, assumed to be high frequency to accommodate increasing traffic	Dynamic and short, ideally every 10 minutes	Not specified, assumed to be dynamic/infrequent yet still cause delays due to long consolidation periods	Several days. A single computation was shown as sufficient for the 15-day period in the evaluation tests	Dynamic. For 10,000 VMs, 30 minutes is required. For 256 VMs, it's 13 seconds	Dynamic and frequent
Ease of Implementation	Physically wiring upper-tier network switches and physical machines which usually have a lot of unused ports. The control logic must be implemented to monitor and react to processors, network traffic, and memory swap statistics	Implemented on NOX - checks traffic to find the pattern and latency. Can be easily applied to currently-deployed or newer network devices	Runs on a test bed & in a simulator that simulates a big data center. It is possible to run CARPO on a network by using SNMP to check the flow of the network and then automatically shut down some switches	Monitors the traffic and changes routing table periodically	Uses VM placement policies that regroup the VMs based on pairing VMs with heavy mutual traffic assigned to host with low cost connection	Checks every paired node and chooses the best one that has the capacity of the overall pairs and with routing build the new network. It is easy to implement
Scalability	Data Centers	Data Centers	Data Centers	Data Centers, ISPs	Data Centers	Data Centers

Embodied Energy Considerations	Implicitly considers embodied energy by creating bypass links to turn off edge switches. Moreover, maximizing the number of VMs on the same server can reduce the number of physical machines. Both solutions can have greater impact on reducing the number of hardware units, thus reducing the embodied energy associated with a given network	Does not take into account embodied energy of servers hosting ElasticTree. Also, it the authors aim to consider "the effect of increasing network size"; "a larger network probably means more, smaller flows, which pack more densely, and reduce the chance of queuing delays and drops". But increasing network size increases embodied energy	Does not take into account the hardware required to implement (8 servers, each with 2 dual-core processors)	No considerations	This method focuses on ideal VM placement so as to reduce the number of physical server - good for embodied energy considerations	No considerations
Redundancy Considerations	Takes VM replica constraints into account when placing them such that a replica is always placed on a different physical machine, even if that compromises energy savings. Uses bypass links if an edge switch is turned off	Adds switches to access level based on the redundancy required. Can minimize the latency time of powering on switches by using sleep mode	None - Possibility of adding constraints during traffic consolidation to guarantee the desired network availability by checking whether shutting down a switch or link would reduce the requested redundancy level	There is a level of redundancy achieved by pre-computing Failover Paths. If they cannot be provided, the algorithm attempts to find the set of paths that are least likely to be all affected by a single failure	Less redundancy than the baseline	Some level of redundancy but not optimal
QoS Considerations	Maintained by satisfying redundancy and creating bypass links. However, the paper does not discuss the possibility of bottlenecks when the algorithm runs to place VMs, reroute traffic, or turn off switches	Responds to sudden increases in traffic, but hindered by the switch boot delay. Learns traffic pattern the system, reserves the required amount of capacity with no latency to meet QoS	There is no compression, but in some cases there is latency in the network, as some paths will be longer	Minimum QoS Delivered, based on the amount of required bandwith	With HOP reduction and load balancing it can meet the requirement for QoS	Monitors traffic real time and reroutes quickly with load balancing. Does not accommodate QoS; quality deteriorates significantly after a certain threshold (400-600 Gbit/s)
Impact on Research Community	Early attempt at energy optimization (2011), but has not been developed further, possibly due to its insignificant energy optimization and use of bypass links, which are ineffective in the case of heavy traffic. Cited by 2 IEEE publications and 13 others	Had a large impact on the community, and is the basis for other methods (e.g., CARPO, Dynamic Traffic, and REsPoNse). Cited by 509 papers on Google Scholar	Has inspired other correlation based traffic engineering and virtualization strategies. It consistently performs well in comparison to them. Cited 65 times	No other strategies were found that utilize pre-computed routing paths. Cited 53 times (mostly for survey purposes of energy optimization methods)	Virtualization is widely employed in network energy optimization. Similar to greedy traffic consolidation, they can introduce bottlenecks and QoS violations. EQVMP addresses that but it does not seem to have an impact on the community (cited on Google Scholar 7 times only for literature review purposes)	Dynamic Traffic is the most recent paper and has only been cited once
Best Environmental Performance Claimed	Up to 7% of network energy	Up to 50% of network energy	Energy savings of up to 46%	Up to 40% energy savings for different traffic matrices	Increases throughput, with some energy savings	Savings of up to 45%
Average Environmental Performance Estimated	1% improvement in the general case	Average energy savings for data centers is 25-40%	41-46%	Average energy savings is 30%	Increases throughput, with some energy savings	Average energy savings is 20%