State-of-the-art Energy Efficiency Approaches in Software Defined Networking

Beakal Gizachew Assefa and Oznur Ozkasap
Department of Computer Engineering
Koc University, Istanbul, Turkey
e-mail: [bassefa13, oozkasap]@ku.edu.tr

***Abstract*—Software Defined Networking (SDN) paradigm has been attracting an increasing research interest. Promising features of SDN are enabling programmable network components and separating the control plane and the forwarding plane. It offers several advantages such as flexibility without sacrificing forwarding performance, high efficiency through optimized routing, ease of implementation and administration, and cost reduction. On the other hand, energy efficiency in networking is an issue as energy cost contributes significantly to the overall costs in information and communication technologies. Thus, energy efficient mechanisms for SDN components have become indispensable. In this study, we address the importance of energy efficiency mechanisms in SDN, propose a classification of methods for improving energy efficiency in SDN and describe each group of solutions with their principles, benefits and drawbacks. To the best of our knowledge, our study is the first one that focuses on state-of-the-art energy efficiency strategies for SDN, discusses open issues and provides guidelines for future research.**

Keywords–Software Defined Networking; SDN; Energy efficiency

I. INTRODUCTION

Software Defined Networking (SDN) is a recent trend in computer networks based on the concepts of control plane and data (forwarding) plane separation and logically centralized control by enabling programmable network devices [1][2]. The idea behind SDN paradigm, depicted in Fig.1, is to eliminate the tight coupling between control and forwarding components in traditional network design, and hence the drawbacks of cumbersome network configuration and limited flexibility to changing requirements. A logically centralized controller configures the forwarding tables (also called flow tables) of switches, which are responsible for forwarding the packets of communication flows.

SDN has been deployed in a diverse set of platforms, ranging from home networks and institutional networks to data center networks. It promises several advantages such as flexibility without sacrificing forwarding performance, high efficiency through optimized routing, ease of implementation and administration, and cost reduction.

The energy consumption constitutes a significant portion of overall information and communication technology costs [3][4]. Several research studies have been conducted for reducing energy costs in different network settings such as wireless sensor networks [5] and cloud data centers [6]. However, to the best of our knowledge, a survey on recent energy saving strategies for SDNs is not available. In this study, we propose a classification of methods for improving energy efficiency in SDN and discuss each group of solutions. We also identify open issues and future research directions.

Energy optimization can be applied at various components of the SDN architecture or SDN itself can be used as a means of energy saving. Energy saving in SDN can be addressed algorithmically or through hardware-based improvements. The hardware-based solutions are applied on the forwarding switches, and such solutions range from compressing the content of Ternary Content Addressable Memory (TCAM) to increasing the capacity of TCAM. Software-based solutions are applied on the controller. We classify state-of-the-art energy efficiency strategies for SDN into four categories, namely traffic aware, compacting TCAM, rule placement, and end host aware.

Traffic aware energy efficiency approaches are inspired by the fact that network components are often under utilized. The key principle is to turn on or off network components (i.e., SDN forwarding switches) based on the traffic load. For instance, when the traffic load is low (e.g., during night time) this approach has the potential to save up to 50% of the total energy consumption [7].

Typically, an elastic tree structure is used to represent the network components that can grow and shrink with the dynamic traffic load. The key challenge is to determine which components to turn off and turn on without compromising the required quality of service (QoS) [7]–[11].

Compacting TCAM solutions attempt to minimize the memory need of information stored in forwarding switches. In SDN, forwarding switches use TCAM, which is a specialized type of high-speed memory that performs an entire memory search in a single clock cycle. However, TCAM is very expensive and power hungry. A memory optimal strategy can be achieved by compacting TCAM itself or compressing the information stored in TCAM [12]–[15].

Rule placement techniques focus on how to place the rules in the forwarding switches. Given the network policies and end point policies, the controller provides a way to convert the high level policies into switch understandable rules. Rule placement is an NP-hard problem which needs a heuristic based solution. Although heuristic based approaches do not guarantee optimal solutions, they typically offer close to optimal results depending on the constraints [16][17][18]. Some of the constraints presented in this area are the maximum number of rules a switch can hold, the routing policy, and the topology. Under such constrains, these approaches attempt to optimize routing.

End host aware energy saving solutions use the practice of turning off underutilized physical servers and running their tasks on a fewer number of servers in SDN based data centers [19][20]. Specifically in data centers, the SDN model is used to form an overlay connecting virtual machines. Server virtualization helps systems to run multiple operating systems and services on a single machine.

The rest of the paper is organized as follows. Section II presents our classification and details of energy efficiency techniques for SDN. In Section III, we discuss open issues on energy efficiency in SDN and provide guidelines for future research. Finally, we give concluding remarks in Section IV.

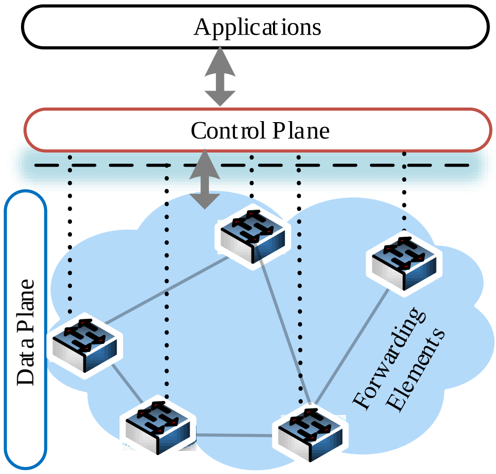

Figure 1: Software Defined Networking Architecture

II. CLASSIFICATION OF ENERGY EFFICIENT SDN SOLUTIONS

In this section, we describe solutions for improving energy efficiency in SDN, based on our classification of techniques, namely traffic aware, compacting TCAM, rule placement, and end host aware. Table I presents a summary of methods under each category along with their key properties.

A. *Traffic Aware Solutions*

With traffic aware energy efficiency approaches, energy consumption can be reduced by turning off some forwarding switches during low traffic load, or putting CPUs or ports at sleep mode. The solutions in this group have the potential to significantly improve energy efficiency in SDN. For data centers, traffic aware approach achieves power savings of up to 50% during low load periods [7].

ElasticTree is a power management solution for data center networks which is implemented on a testbed consisting of OpenFlow switches [7]. The idea is to turn off links and switches based on the amount of traffic load. Therefore, energy consumption of the network is proportional to the dynamically changing traffic.

ElasticTree consists of three optimizers: formal model, greedy bin-packing, and a topology-aware heuristic. Each optimizer takes network topology (a graph), routing constraints, power model (flat), and traffic matrix as input, and outputs subset of links and flow routes.

The formal model formulates the power saving problem by specifying objective function and constraints. The objective function minimizes the sum of the total number of switches turned on and the number of links. The advantage of the formal model is that it guarantees a solution within some configurable optimum; however, it only scales up to 1000 hosts.

The greedy bin-packing optimizer evaluates possible paths and chooses in left-to-right order manner that is the left most path is chosen first. The optimizer improves the scalability of the formal model. This approach suffers the same problem as any of heuristic techniques. However, solutions can be computed incrementally and can support on-line usage.

The topology-aware heuristic optimizer, on the other hand, splits the flow and finds the link subset easily. It is computationally efficient, since it takes advantage of a fat tree structure and takes only port counters to compute link subset. This approach uses IP to formalize the optimization problem. The drawback is degradation of performance because of turning on and turning off components.

CoRrelation-aware Power Optimization(CARPO) algorithm dynamically consolidates traffic flows onto a small set of links and switches in a data center network, and then shuts down unused network devices for energy savings [8]. It consolidates traffic flows based on correlation analysis among flows. Another important feature of CARPO is to integrate correlation-aware traffic consolidation with link rate adaptation for maximized energy savings. The integration is formulated as an optimal flow assignment problem. A near-optimal solution is first computed using linear programming to determine consolidation and the data rate of each link in the data center network. A heuristic algorithm is used to find a consolidation and rate configuration solution with acceptable runtime overheads. The heuristic reduces the computation complexity.

REsPoNse is a framework that allows network operators to automatically identify energy-critical paths [9]. It investigates the possibility to pre-compute a few energy-critical paths that, when used in an energy-aware fashion, can continuously produce close-to-optimal energy savings over long periods of time. REsPoNse identifies energy-critical paths by analyzing the traffic matrices, installs them into a small number of routing tables (called always-on, on-demand, and fail-over), and uses a simple, scalable online traffic engineering mechanism to deactivate and activate network elements on demand. The network operators can use REsPoNse to overcome power delivery limits by provisioning power and cooling of their network equipment for the typical, low to medium level of traffic.

A similar technique named Carrier Grade is proposed in [10]. While the Openflow architecture may not be able to significantly reduce energy consumption by consolidating the control hardware/software in a single machine, it shows significant promise by facilitating network wide energy efficiency solutions. MLTE was implemented in combination with local

TABLE I: SUMMARY OF ENERGY EFFICIENCY TECHNIQUES IN SDN

Category	Approach	Properties
Traffic aware	ElasticTree [7]	ElasticTree, based # traffic, Mixed IP, re-computation cost
	CARPO [8]	Fat Tree, based on correlation analysis among flows
	REsPoNse [9]	FatTree, identify energy-critical path to optimize
	Carrier Grade [10]	MLTE implementation, topology-aware heuristic
	Integrated [11]	Combined sleep and turning off, recovery from failure
Compacting TCAM	Rectilinear [12]	Rectangle Rule List (RRL) minimization, geometric model
	TCAM Razor [13]	Decision diagrams, dynamic programming, and redundancy removal
	Bit Weaving [14]	Non-prefix based compression
	Compact TCAM [15]	Usage of short tag
Rule Placement	Big Switch [17]	Heuristics for endpoint policy, routing policy and rule placement
	Palette [16]	Graphs, algorithms, heuristics
	Optimizing Rule Placement [18]	Meaning of rules, integer linear programming formulation of the problem
End host aware	Honeyguide [19]	VM migration, fault tolerance, easy deployment
	EQVMP [20]	Virtual Machine, load balancing

power saving options such as controlled adaptive line rates in the Openflow switches [21]. The technique is also extended to handle failures which happen in the controller or forwarding switches.

An integrated scheme that combines smart sleeping and power scaling based on the topology-aware heuristic algorithm to improve energy saving level of data center networks is proposed in [11]. This combined mechanism was deployed in a data center using Fat-Tree topology, and the bounds on energy savings in low and high traffic utilization cases were analysed. Analytical results show that the combined algorithm reduces energy consumption remarkably compared to the conventional one in case of high traffic.

B. Compacting TCAM Solutions

Content Addressable Memory (CAM) is a special type of memory that enables direct query to the content without having to specify its address. A search in CAM provides a search tag which is the representation or the content itself, and returns the address of the content if the item is found. The content is represented in binary (i.e., 0 and 1). TCAM is a specialized CAM, and the term ternary refers to the memory's ability to store and query data using three different inputs: 0, 1 and X. The X input, which is often referred to as a wildcard state, enables TCAM to perform broader searches based on pattern matching. TCAM is popular in SDN switches for fast routing lookup and it is much faster than RAM. However, TCAM is expensive, consumes high amount of power, and generates a high level of heat. For example, a 1Mb TCAM chip consumes 15-30 watts of power, and TCAM is at least as power hungry as SDRAM. Power consumption together with the consequent heat generation is a serious problem for core routers and other networking devices [13][22].

Two kinds of compression that can be applied in TCAM are rule compression and content compression. In a traditional Access Control List, a rule has five components: source range, destination range, protocol, port(s), and action. In SDN, the forwarding decision of a switch is based on flow tables implemented in TCAM. Each entry in the flow table defines a matching rule and is associated with an action. Upon receiving a packet, a switch identifies the highest-priority rule with a matching predicate, and performs the corresponding action. The proposals in [12]-[15] are attempts to compact these rules to utilize TCAM effectively.

Rectilinear [12] is an approach that exploits SDNs features such as programming interface to the switches and dynamic determination of actions for each flow at the switches. The compacting reduces the size of bits to store information that are essential to classify packets to a flow. A flow-id is given to each flow to uniquely identify packets in the corresponding flow. The packet headers are modified at the forwarding switches to carry the flow-id that can be used by other switches on the path for classifying the packets. A shorter tag representation for identifying flows than the original number of bits are used to store the flow entries of SDN switches. The authors demonstrated that the compact representation on flow can reduce 80% TCAM power consumption on average.

TCAM Razor proposes a four step solution to compress the packet classifier [13]. First, it converts a given packet classifier to a reduced decision diagram. Second, for every non-terminal node in the decision diagram, it minimize the number of prefixes associated with its outgoing edges using dynamic programming. Third, it generates rules from the decision diagram. Last, it removes redundant rules.

The Bit Weaving technique employs a non-prefix compression scheme [14], and is based on the observation that TCAM entries that have the same decision but whose predicates differ by only one bit can be merged into one entry by replacing the bit in question with *. Bit weaving consists of two new techniques, bit swapping and bit merging. First it swaps bits to make the similar and then merge such rules together. The key advantages of bit weaving are that it runs fast, it is effective, and can be complementary to other TCAM optimization methods as a pre/post-processing routine.

A Compact TCAM approach that reduces the size of the flow entries is proposed in [15] that uses shorter tags for identifying flows than the original number of bits used to store the flow entries for SDN switches. The catch for this approach comes from the dynamic programming capability of SDN to route the packets using these tags. Furthermore,the usage of SDN framework to optimize the TCAM space is introduced.

C. Rule Placement Solutions

The energy efficiency techniques for rule placement solutions start by formalizing the energy cost model and the constraints associated, then applies heuristic technique to find optimum energy saving strategy. Forwarding rules are generated and pushed to the forwarding switches by the controller. The controller generates rules and pushes them to the forwarding switches. Placing the rules to respective switches distributed across the network and optimizing given an objective function under the constrains is NP-hard problem. The objective function in our particular case is minimizing energy where as the constrains are the the number of switches, flow table capacity, link capacity, and number of ports per switch. SDN makes use of a logically centralized controller which has a global view of the network that provides flexibility for optimizing forwarding routes.

The Palette distribution framework is a distributed approach applied to SDN tables [16]. Since the SDN controller table can only handle hundreds of entries, and the memory is expensive and power hungry, Palette decomposes large SDN tables into small ones and then distributes them across the network, while preserving the overall SDN policy semantics. Palette helps balance the sizes of the tables across the network, as well as reduce the total number of entries by sharing resources among different connections. It copes with two NP-hard optimization problems: decomposing a large SDN table into equivalent sub-tables, and distributing the sub-tables. The problem of traversing is formulated using the rain-bow problem.By giving unique color for each sub-tree, each connection traverses each color type at least once. Implementation of Palette is based on graph theory formulation algorithms and heuristics.

Big Switch approach utilizes the fact that SDN controllers have a global view of the network and proposes that the entire network should be viewed as one big switch [17]. The architecture modularizes the SDN controller into three components: endpoint policy, routing policy and rule placement policy. A high-level SDN application defines end-point connectivity policy on top of big switch abstraction; a mid level SDN infrastructure layer should decide on the routing policy; and a compiler should synthesize on the end-point and routing policy and develop an effective set of forwarding rules that obey the user-defined policies and adhere to the resource constraints of the underlying hardware. Minimizing the number of rules needed to realize the endpoint policy under rule capacity constraint is both a decision and an optimization problem. The architecture addresses the two problems through a heuristic algorithm that recursively covers the rules and packs groups of rules into switches along the path.

The drawbacks of Pallete and Big Switch approaches are that they do not rely on the exact meaning of the rules and the rules should not determine the routing of the packets.A technique of compacting rules, which enhances the rule placement, is proposed in [18]. This approach analizes the meaning of the rules, and together with heuristic optimization method to minimize energy consumption for a backbone network while respecting capacity constraints on links and rule space constraints on routers. They present an exact formulation using integer linear programming, and introduce efficient greedy heuristic algorithm for large networks.

D. End Host Aware Solutions

Server virtualization enables the working of multiple virtual machines simultaneously on a single physical server, thus decreasing electricity consumption and wasting heat as compared to running underutilized physical servers. Hence, instead of operating many servers at low utilization, virtualization technique combines the processing power onto fewer servers that operate at a higher total utilization. The deployment of SDN in cloud data center virtual machines boosts QoS and load balancing. Unlike traffic aware strategies, where the network components are the focus for energy saving, the virtual machines are utilized in the saving strategy.

Honeyguide is a virtual machine migration-aware network topology for energy efficiency in data center networks [19]. Reducing energy consumption is achieved by decreasing the number of active (turned on) networking switches. In this approach, the focus is not only turning off inactive switches, but also trying to maximize the number of inactive switches. To increase the number of inactive switches, two techniques are combined: virtual machine (VM) and traffic consolidation. As an extension of existing tree based topologies, Honeyguide adds bypass links between the upper-tier switches and physical machines. By doing so, it meets the fault tolerance requirement of data centers. It is easily deployable since what it needs is to add a bypass link only.

EQVMP proposes energy-efficient and QoS-aware virtual machine placement for SDN based data centers [20]. Unlike ElasticTree, power on and off is applied to the servers themselves. EQVMP combines three techniques: hop reduction, energy saving and load balancing. Hop reduction divides the VMs into groups and reduces the traffic load among groups by graph partitioning. Energy savings mostly are achieved by VM placement. The motivation behind VM placement is from Best Fit Decreasing (BFD) and Max-Min Multidimensional Stochastic Bin Packing. Fat-tree is used to represent the VM and servers in the data center. SDN controller is used to balance the load in the network. Load balancing achieves flow transmission in networks without congestion.

III. OPEN ISSUES AND DISCUSSION

For energy efficiency techniques in SDN, we identify the following open issues and future research directions.

- In traffic aware solutions, turning the network components on and off based on network load helps in reducing energy consumption. However, determining the set of network components to turn on or turn off dynamically without affecting QoS and performance is an NP-hard problem. An efficient solution in this area should consider the trade off between energy savings and network performance.
- Other open research issues in traffic aware energy efficiency are scalability and flexibility. The traffic aware

model needs to be scalable in the case of high traffic load. Flexibility is the ability of the system to adapt dynamic network settings (i.e., different topologies, change in the number of nodes).

- Forwarding switches use TCAM, which is expensive and power hungry. Some research proposals have attempted to compact the rules stored in TCAM. For such compacting TCAM solutions, information stored in TCAM cannot be further compressed after a certain threshold.
- Formal definition of the energy saving problem is the base for applying a sound theoretical solutions. Since the problem is NP-hard, utilizing heuristic techniques is inevitable. On an average case, heuristics can give close to optimal solution with in a feasible time. Specially, in a dynamic environment, the problem becomes even more challenging.
- Rule placement directly affects both the performance of the network and also determines the routing. Given a routing policy and end policy of a network, there is a need for formal and more space efficient way of representing rules.
- There have been few studies in end host aware energy efficiency strategies. Combining the advantages of virtualization with SDN can improve performance and minimize energy consumption.

IV. CONCLUSION

In this paper, we address the importance of energy efficiency mechanisms in software defined networks. With the significance of energy efficiency in networking, mechanisms enabling energy savings in the SDN model become indispensable. We propose a classification of methods for improving energy efficiency in SDN (traffic aware, compacting TCAM, rule placement, and end host aware) and discuss each group of solutions. To the best of our knowledge, our study is the first one that focuses on recent energy saving strategies for SDN. As future work, we aim to conduct a comprehensive study on the methods, make a experimental comparisons, and develop measurement metrics.

Acknowledgements

This work was partially supported by the COST (European Cooperation in Science and Technology) framework, under Action IC0804 (Energy Efficiency in Large Scale Distributed Systems), and by TUBITAK (The Scientific and Technical Research Council of Turkey) under Grant 109M761.

REFERENCES

[1] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, “A survey of software-defined networking: Past, present, and future of programmable networks,” Communications Surveys Tutorials, IEEE, vol. 16, no. 3, 2014, pp. 1617–1634.

[2] K. Diego et al., “Software-Defined Networking: A Comprehensive Survey,” Proceedings of the IEEE, vol. 103, no. 1, Jan 2015, pp. 14–76.

[3] L. Chiaraviglio, M. Mellia, and F. Neri, “Minimizing isp network energy cost: Formulation and solutions,” IEEE/ACM Transactions on Networking, vol. 20, no. 2, April 2012, pp. 463–476.

[4] M. Pickavet et al., “Worldwide energy needs for ict: The rise of power-aware networking,” in Advanced Networks and Telecommunication Systems (ANTS), 2008. 2nd International Symposium on, Dec, pp. 1–3.

[5] R. Soua and P. Minet, “A survey on energy efficient techniques in wireless sensor networks,” in Wireless and Mobile Networking Conference (WMNC), 2011, 4th Joint IFIP, Oct, pp. 1–9.

[6] A. Hammadi and L. Mhamdi, “A survey on architectures and energy efficiency in data center networks,” Computer Communications, vol. 40, 2014, pp. 1 – 21.

[7] B. Heller et al., “Elastictree: Saving energy in data center networks,” in NSDI, 2010.

[8] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, “Carpo: Correlation-AwaRe power optimization in data center networks,” in IEEE INFOCOM, March 25-30, 2012, pp. 1125–1133.

[9] N. Vasić et al., “Identifying and using energy-critical paths,” in Seventh Conference on Emerging Networking EXperiments and Technologies, ser. CoNEXT, 2011. ACM, pp. 18:1–18:12.

[10] D. Staessens, S. Sharma, D. Colle, M. Pickavet, and P. Demeester, “Software defined networking: Meeting carrier grade requirements,” in Local Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop on, Oct, pp. 1–6.

[11] T. Nguyen et al., “Modeling and experimenting combined Smart sleep and power scaling algorithms in energy-aware data center networks,” Simulation Modelling Practice and Theory, vol. 39, 2013, pp. 20 – 40.

[12] D. A. Applegate et al., “Compressing rectilinear pictures and minimizing access control lists,” in ACM-SIAM Symposium on Discrete Algorithms (SODA), 2007, pp. 1066–1075.

[13] C. R. Meiners, A. X. Liu, and E. Torng, “Tcam razor: A systematic approach towards minimizing packet classifiers in tcams,” in 15th IEEE International Conference on Network Protocols (ICNP), Beijing, China, October 2007.

[14] C. Meiners, A. Liu, and E. Torng, “Bit weaving: A non-prefix approach to compressing packet classifiers in tcams,” IEEE/ACM Transactions on Networking, vol. 20, no. 2, April 2012, pp. 488–500.

[15] K. Kannan and S. Banerjee, “Compact TCAM: flow entry compaction in TCAM for power aware SDN,” IEEE/ACM Transactions on Networking, vol. 20, no. 2, Apr. 2012, pp. 488–500.

[16] Y. Kanizo, D. Hay, and I. Keslassy, “Palette: Distributing tables in software-defined networks,” in IEEE INFOCOM, April 2013, pp. 545–549.

[17] N. Kang, Z. Liu, J. Rexford, and D. Walker, “Optimizing the ”one big switch” abstraction in software-defined networks,” in Ninth ACM Conference on Emerging Networking Experiments and Technologies, ser. CoNEXT. New York, NY, USA: ACM, 2013, pp. 13–24.

[18] F. Giroire, J. Moulierac, and T. K. Phan, “Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing,” Tech. Rep., 2014.

[19] H. Shirayanagi, H. Yamada, and K. Kono, “Honeyguide: A VM migration-aware network topology for saving energy consumption in data center networks,” in Computers and Communications (ISCC), IEEE Symposium on, July 2012, pp. 460–467.

[20] S.-H. Wang, P.-W. Huang, C.-P. Wen, and L.-C. Wang, “EQVMP: Energy-efficient and qos-aware virtual machine placement for software defined datacenter networks,” in Information Networking (ICOIN), International Conference on, Feb 2014, pp. 220–225.

[21] B. Puype et al., “Multilayer traffic engineering for energy efficiency,” Photonic Netw. Commun., vol. 21, no. 2, Apr. 2011, pp. 127–140.

[22] W. Jiang, “Scalable ternary content addressable memory implementation using fpgas,” in Architectures for Networking and Communications Systems (ANCS), ACM/IEEE Symposium on, Oct 2013, pp. 71–82.