Green DataPath for TCAM-Based Software-Defined Networks

Huawei Huang, Song Guo, Jinsong Wu, and Jie Li

A TCAM-based flow table is a power-hungry hardware that can provide high-speed lookup operations for packet switching networks. A number of energy-efficient TCAM usage approaches have been proposed, but this topic still has not been well studied in the context of traffic engineering for SDN. Aiming to find energy-efficient routing paths for traffic sessions in SDN networks, the authors propose a novel Green DataPath architecture, where the dynamic voltage and frequency scaling power management technique is introduced.

Abstract

A TCAM-based flow table is power-hungry hardware that can provide high-speed lookup operations for packet switching networks. A number of energy-efficient TCAM usage approaches have been proposed in recent literature, but this topic still has not been well studied in the context of traffic engineering for SDN. To this end, aiming to find energy-efficient routing paths for traffic sessions in SDN networks, we propose a novel Green DataPath architecture, where the dynamic voltage and frequency scaling (DVFS) power management technique is introduced. The dedicated SDN controlling module and DVFS-enabled switches are devised for the control plane and data plane, respectively. A framework for energy-efficient routing algorithms is also developed under the proposed architecture and evaluated by extensive simulations using three traffic scheduling schemes.

Introduction

In the last decade, ternary content addressable memory (TCAM) has beceme the dominant hardware, providing super high-speed forwarding operation in packet switching networks. For example, a commercial TCAM chip named R8A20410BG can support 20 Mbits density working at 360 MHz per table, which suggests that it can perform up to 360 million searches per second per table. While a TCAM has line-rate speed lookup benefits, it also comes with disadvantages such as the high cost-to-density ratio (US$350 for a 1 Mbit chip) and high power consumption (15–30 W/Mb).

Due to these reasons, TCAMs have been limited to wild card storage in packet switching devices, and must be carefully scheduled to use. Therefore, a number of energy-efficient TCAM usage approaches have been proposed for packet switching networks in the state-of-the-art literature. These approaches can be classified into three categories: *TCAM usage reduction* [1, 2], *TCAM partial utilization* [3–5], and *forwarding rule compression* [6–10], while they are still in the premature stage in the context of traffic engineering for software defined networking (SDN).

In this article, we call the traffic flows between the data source transmitter and the client receiver a *session*. To find the energy-efficient routing paths for traffic sessions in SDN networks, we present a novel Green DataPath architecture, where the dynamic voltage and frequency scaling (DVFS) power management technique is used. In particular, we have also devised the dedicated SDN controlling logic and DVFS-enabled switches. Under the Green DataPath architecture, a routing algorithm is proposed and evaluated under various network settings.

Preliminaries

SDN

SDN simplifies network management via decoupling the control and data planes using a logically centralized network operating system called a controller. Thus, complicated controlling logics are no longer necessarily installed in packet forwarding devices such as switches or routers. Therefore, SDN has been viewed as the next generation network paradigm [11]. In SDN networks, each SDN switch at the data plane conducts data forwarding according to the flow table entries (also called rules) installed by the controller. Each forwarding rule can be expressed in the form of ⟨*Match*, *Action*⟩, in which the *Match* field is used to match against the packet header. If a rule is matched, the switch executes the specified actions in the *Action* field to the packet. For example, the rule ⟨*Match*:{*ip*, *nw_src* = 100.0.0.1, *nw_dst* = 100.0.0.2}, *Action*=*output*:3⟩ indicates that the packets from a host with source IP address 100.0.0.1 and a destination IP address 100.0.0.2 will be forwarded to the *output* port 3 of the switch.

TCAM

In Ethernet networks, switches and routers must deliver bandwidth-hungry services such as voice over Internet Protocol (VoIP), IP television (IPTV), video on demand (VOD), and wireless third/fourth generation (3G/4G) with the appropriate quality of service (QoS) levels. In order to build the platforms necessary to optimally manage large amounts of network traffic quickly and effectively, system designers are increasingly relying on advanced content addressable memory (CAM), especially TCAM devices, to perform ultra-fast data packet searches.

CAM compares input search words, such as the match fields in packet headers, against a table of stored forwarding rules, and returns the address of the matched data. CAM can finish a complete lookup operation over all stored rules

Digital Object Identifier: 10.1109/MCOM.2016.1600067CM

Huawei Huang and Song Guo are with the University of Aizu; Jinsong Wu is with Universidad de Chile; Jie Li is with the University of Tsukuba.

0163-6804/16/$25.00 © 2016 IEEE

in a single clock cycle. Therefore, it is popular in high throughput systems. Figure 1a illustrates an example of the lookup operation. When a packet with the source IP address 100.0.0.1 arrives at a switch, the packet header will be compared against the rule prefixes stored in `CAM based table`. The matched prefix, such as the shadowed one, will activate the corresponding matchline, which generates an encoding signal. After decoding such a mapping signal by `Decoder`, the predefined action, such as 100.0.0.1:Output 3, will be duplicated to the `Action execution` module. Finally, the processed packet leaves the current switch.

In general, there are two types of CAMs: binary CAM (BCAM) and TCAM. The former can be used to store full entries and perform exact 0/1 lookup against each bit of the search data, while the latter can store wildcard entries and do more beyond the binary comparison. In each wildcard entry, the "X" value, called a "*don't care*" bit, can also be represented, indicating that a particular bit in the search data will not be taken into consideration when comparing with a stored rule. This feature is very useful in many applications such as the prefix matching in IP-lookup and range queries for packet classification. In order to support three states of each bit in a rule, that is, *match 0*, *match 1*, and *don't care*, each TCAM cell requires encoding using two physical bits. For example, Fig. 1b illustrates a NOR-type-based TCAM cell, which contains two static random access memory (SRAM) cells representing two physical bits, D_0 and D_1. Since each physical bit can represent two binary states, the combination of D_0 and D_1 can denote four logical possible states, but only three of them are required by the ternary storage. On the other hand, Fig. 1c shows the ternary encoding table for the NOR-type-based TCAM cell, where we set $D_0 = 0$, $D_1 = 1$, and $D_0 = 1$, $D_1 = 0$ to store logic ternary symbols "0" and "1," respectively. Additionally, the cell allows searching for an "X" symbol by setting both SL0 and SL1 to logic "0." This is an external "*don't care*" that forces a match of a bit regardless of the stored bit. Therefore, using TCAM, a packet forwarding device can do *wildcard* lookup operations.

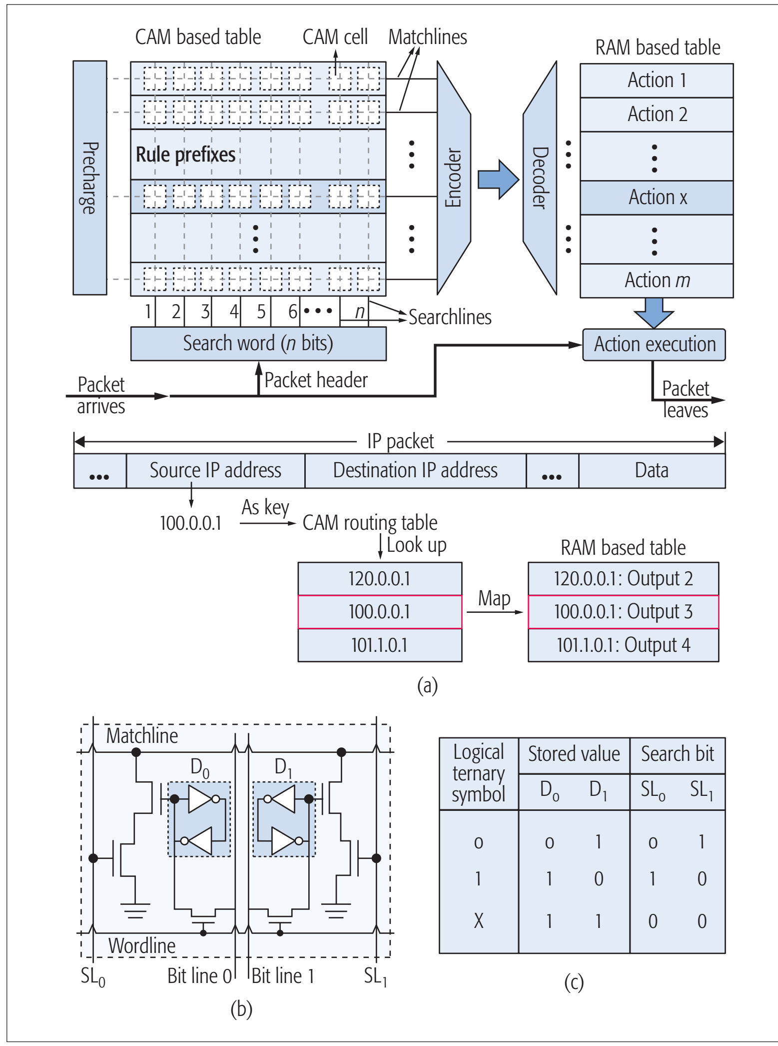
In the early stage of IP routers, the lookup speed was unable to match the growth of link bandwidth. TCAMs have been adopted to design high throughput forwarding engines on routers and switches [6].

Due to the realization of the logical ternary symbol, TCAMs are more expensive and consume much more circuit board space than SRAMs. In the fast lookup operation, TCAM chips also generate a large amount of heat. Therefore, a TCAM cell is far more complicated and power-consuming than an SRAM cell. For example, 1 Mbit TCAMs consume 15–30 W of power, about 50 times higher than SRAMs.

Power Consumption of TCAM

The power consumption of a hardware component mainly depends on the voltage supply (V_{dd}), equivalent capacitance (C_{eq}), and operating frequency (f). Most of the power consumption in TCAM is due to charging and discharging of various control lines, such as the searchlines and matchlines shown in Fig. 1a. According to the TCAM power consumption model presented in [12], $P \propto \cdot V2_{dd} \cdot f$, we can infer that three parameters (i.e., V_{dd}, C_{eq}, and f) can be tuned directly to reduce power consumption. Based on the impact of parameter variations on circuits and micro-architecture [13], the following insights of parameter configuration are obtained regarding power reduction on a transistor:
- The capacitance of transistors in an off-the-shelf TCAM chip is already ossified and cannot be tuned.
- The variation of supply voltage takes a significant percentage of the entire supply voltage range. Too high voltage leads to unreliable processing upon packets, while too low voltage increases the probability of failure during read and write operations.
- Too low frequency leads to low performance (e.g., low processing speed), and too high frequency results in high leakage power. Note that a wide spread of reasonable frequency distribution does exist.

In summary, the reduction of power consumption can be achieved by reasonably tuning V_{dd} and f.



Logical ternary symbol	Stored value D_0	Stored value D_1	Search bit SL_0	Search bit SL_1
0	0	1	0	1
1	1	0	1	0
X	1	1	0	0

Figure 1. The rationale of CAM-based lookup operation and TCAM cell: a) rationale of packet lookup operation in a switch/router; b) a NOR-type TCAM cell; c) ternary encoding for a NOR cell.

Once a dynamic traffic flow arrives, the expense of refreshing the current solution is significantly high or even intractable. Therefore, designing energy-efficient strategies that can handle the highly dynamic arriving flows in an online fashion is a critical challenge.

Category	Literature	Critical component(s)	Power awareness	Rule compression ratio	Dynamic rule update
TCAM usage reduction	Yamanaka [1]	Match field translator	No	Not applicable (N/A)	Good
	Congdon [2]	Signature CAM, prediction circuitry	Yes	(N/A)	Poor
TCAM partial utilization	Panigrahy [3]	ASIC based prefix indexer	Yes	(N/A)	Poor
	CoolCAMs [4]	Bit-selection logic	Yes	(N/A)	(N/A)
	Ma [5]	CAM based pre-classifier	Yes	(N/A)	Fair
Rule compression	EaseCAM [6]	Prefix aggregation and expansion techniques	Yes	Fair	Fair
	Meiners [7]	TCAM Razor approach	No	High	Poor
	Sun [8]	Tree representation	No	High	Poor
	Compact TCAM [9]	Shorter tags	Yes	Fair	Fair
	Multiplexer [10]	Rule-multiplexing scheme	No	High	Fair
DVFS-based	This article	Chip-indexer, DVFS module	Yes	(N/A)	Good

Table 1. Comparisons on the energy-efficient TCAM usage.

State-of-the-Art Energy-Efficient TCAM Usage

When applying energy-efficient lookup operation in physical packet-switching networks, recent related works can be generally classified into the three aforementioned categories. The remarkable properties of these existing proposals are summarized in Table 1.

Taxonomy of Energy-Efficient TCAM Usage

Category-A: TCAM Usage Reduction: In order to offload TCAM usage, Yamanaka *et al.* [1] built a "matching field translator" architecture, in which a list of exact match rules are generated for a corresponding wildcard rule in the first step, and then a controller translates exact matching fields into the source medium access control (MAC) addresses based on the correspondence between them. As a result, only the shorter rules that contain MAC addresses are necessary and can be stored in BCAM of a switch. Similarly, Congdon *et al.* [2] created a signature CAM and RAM-based packet parser, which works as prediction circuitry. According to the prediction logic results (i.e., prediction hit, incorrect prediction, and prediction miss), the TCAM utilization manners are attributed to no-TCAM, only using master-TCAM and full-TCAM usage, respectively.

Category-B: TCAM Partial Utilization: Panigrahy *et al.* [3] partitioned TCAMs into several groups first, and then used an application-specific integrated circuit (ASIC)-based hash table to perform lookup in only one TCAM chip, others remaining inactive. Similarly, Zane *et al.* [4] proposed a bit-selection logic to reduce power consumption. In the proposed architecture, TCAMs are partitioned to different blocks, and the hashing bits are selected to point to specified TCAM subtables. Recently, Ma *et al.* [5] introduced a smart pre-classifier which classifies a packet in advance such that only a small portion of TCAM will be activated and searched for a given packet.

Category-C: Forwarding Rule Compression: Ravikumar *et al*. [6] introduced prefix aggregation and expansion techniques, aiming to activate a limited number of TCAM arrays during IP lookup. In such a way, the effective TCAM size in a router can be compacted. To address the range expansion problem of TCAM installation, Meiners *et al.* [7] considered how to generate a semantically equivalent packet classifier that requires the minimum number of rules for a given set of original TCAM entries. Using tree representation of rules, Sun et al., [8] proposed a redundancy removal algorithm, which removes redundant rules and combines overlapping rules to build an equivalent and smaller rule set for a given packet classifier. Kannan *et al.* [9] used shorter tags for identifying flows than the original ones used to store the flow entries. As a result, the size of forwarding rules can be reduced. In order to efficiently use TCAM space, a rule multiplexing scheme in [10] was proposed with joint optimization on traffic engineering in SDN networks. Using this scheme, the original same set of rules deployed on each node for the whole flow of a session but toward different paths can be compacted in some particular overlapped switch nodes, such that the occupied TCAM space is reduced.

Challenges

Update Rules for Dynamic Traffic Flows: As observed from the related work presented above, most existing approaches can only be operated offline.

For example, in the approaches belonging to Category-B, ASIC-based hash tables were used

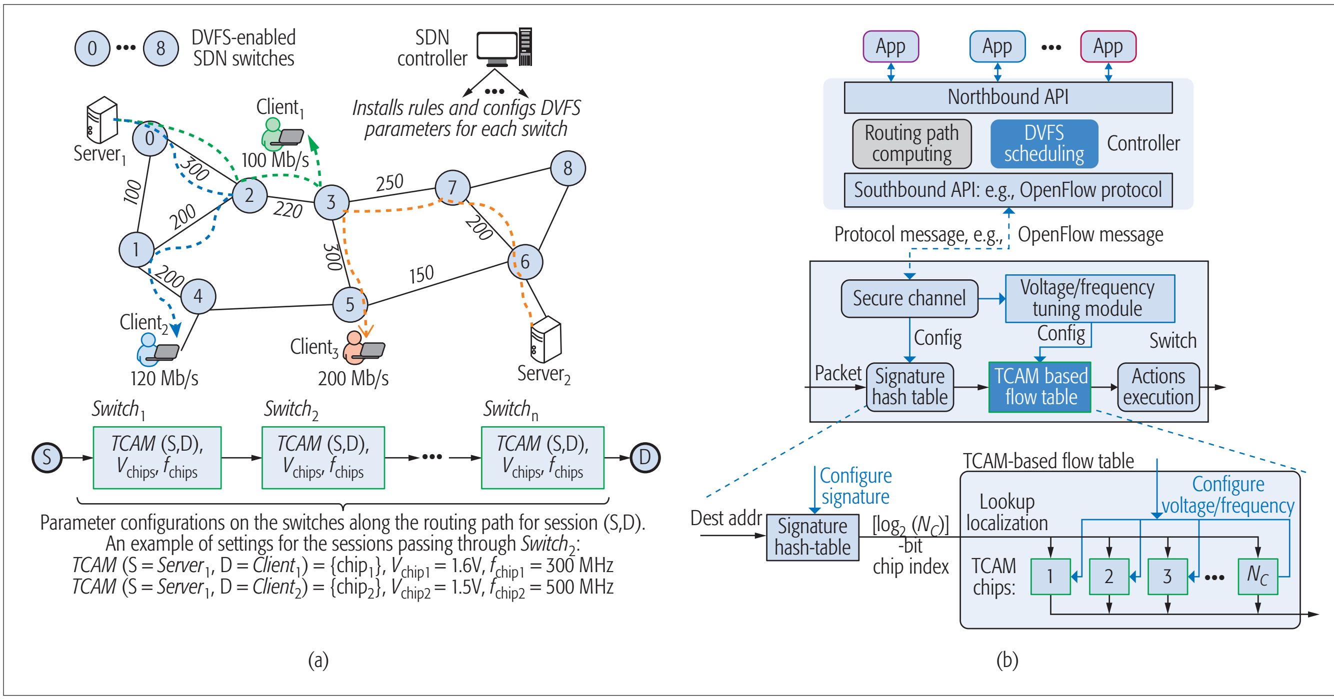


Figure 2. Green DataPath architecture: a) a use case under Green DataPath architecture; b) DVFS-enabled controller/switch.

to map the arriving packets to the specified TCAM chips [3, 4], and the indexing entries were used in a TCAM based pre-classifier [5]. However, how to update these hash-table/pre-classifier entries when new traffic flows arrive still has not been well resolved.

In practice, traffic flows usually arrive at a switch randomly and dynamically, resulting in the controller having to install time-varying rule sets on the traversed switches. Once a dynamic traffic flow arrives, the expense of refreshing the current solution is significantly high or even intractable. Therefore, designing energy-efficient strategies that can handle the highly dynamic arriving flows in an online fashion is a critical challenge.

QoS Guarantee: The other challenge is how to guarantee the QoS requirement of client flows while saving energy.

Most existing approaches enforce additional processing before lookup operations using TCAMs. For example, the method proposed in [1] requires rule set translation. Moreover, all proposals in Category-C have to execute complicated algorithms to compact the original rule set into a smaller one. All such pre-lookup operations introduce a non-negligible delay and potentially degrade the QoS performance of client flows.

Green DataPath Architecture

In this section, we present a novel Green DataPath architecture, which is a totally different approach in the context of traffic engineering for SDNs compared to the earlier reviewed three categories. The reason is that the DVFS technique [14] is particularly introduced to the proposed architecture.

Overview

We first present an overview of the Green DataPath Architecture by displaying a use case demonstrated in Fig. 2a. We consider an SDN network with a centralized controller and several DVFS-enabled SDN switches/routers, in which three sessions are being served under the Green DataPath architecture. When a client S requests an app service from data source D, the controller decides a path for session (S, D) along with a set of rules installed on the traversed switches. Particularly, it also specifies dynamic frequency and voltage parameter configurations for the TCAMs in each switch. When a switch receives a DVFS parameter configuration message, its embedded TCAMs shall work in the specified state through a voltage/frequency tuning module.

When the controller installs forwarding rules into the destined TCAM chip of each passing switch during a routing path, the operating frequency and voltage of the target TCAM chip need to be specified, even if this TCAM chip is empty (there are no rules being stored). The bottom figure in Fig. 2a illustrates an example of parameter configurations for two sessions, in which for the $Client_1$ oriented session, the parameter configurations toward $Switch_2$ are given as TCAM ($S = Server_1$, $D = Client_1$): $\{chip_1\}$, $f_{chip_1} = 300$ MHz, $V_{chip_1} = 1.6$ V. Such a configuration indicates that the forwarding rule for this session decides to install on TCAM $chip_1$, with an operating frequency of 300 million Hz and an operating voltage of 1.6 V. Note that TCAM could perform one packet search per cycle clock per table/chip. That is, the corresponding search speed is 300 million searches per second (MSPS) in TCAM $chip_1$. The frequency/voltage configuration of a TCAM chip can be tuned adaptively. Specifically, the frequency of a TCAM chip can be increased without exceeding the reliable threshold when new traffic sessions pass through. Therefore, which TCAM chip should be chosen to place the rules for each session is also a critical problem in the Green DataPath architecture.

If a TCAM chip is being installed with rules, we call it an activated chip. All activated TCAM chips are working under the scheduled voltage/frequency settings. If a match is hit, the corresponding action will be executed to the target packet, which is finally delivered out of the switch. Otherwise, the packet may be dropped or sent to a controller for further consultation.

DVFS-Enabled Controller

In order to support our proposed architecture, a dedicated controlling logic and a DVFS tuning module need to be supported by the controller and switch, respectively. Figure 2b displays the design of the DVFS-enabled controller and switch. The logically centralized controller governs the global information of the entire network such as the connectivity map, link bandwidth, and TCAM chip occupation status. Furthermore, it implements the DVFS mechanism through the following two major modules shown in Fig. 2b.

•The routing path computing module can be implemented by a path searching algorithm, for example, Dijkstra's shortest path algorithm, aiming to find a routing path for traffic flows based on the global network overview.

•The DVFS scheduling module is responsible for generating the frequency and voltage configurations according to a certain energy efficiency policy for TCAM chips. When the first packet that indicates a new traffic session arrives at the controller via a protocol message such as the OpenFlow *packet_in* message, another function of the DVFS scheduling module is to put a unique binary signature in the specific header field of this packet and send it to the ingress switch via a protocol message like the OpenFlow *packet_out* message.

It is worth noting that a unique signature witha $\lceil \log_2(N_C) \rceil$-bit, where N_C is the number of TCAM chips in a switch, is used to specify the target TCAM chip and can be parsed by a switch. All the packets belonging to a specific session will be directed to the indexed TCAM chip for lookup processing. We call such a pre-classification the *lookup localization*.

DVFS-Enabled Switch

On the other hand, DVFS-enabled switches conduct data forwarding according to the rules installed by the controller. Corresponding to the critical components, that is, the *Signature hash-table* and *voltage/frequency tuning module*, which are shown in the bottom of Fig. 2b, we describe the relevant operations as follows.

- The forwarding rules, packet signature, as well as the voltage/frequency configurations are delivered from the controller to switches via protocol messages (e.g., the OpenFlow *packet_out* messages), and finally arrive at the *secure channel* which resides in a switch.
- The *signature hash table* stores the TCAM chip indexed hash table entries, each of which consists of the destination address prefix and $\lceil \log_2(N_C) \rceil$-bit signature. The hash table is used to perform the *lookup localization* discussed above.
- The *voltage/frequency tuning module* is designed to assign the voltage and frequency setting for TCAM chips.
- The received rules are cached in the specified TCAM chips.
- Finally, after lookup operation, packets are processed by the *Actions execution* module.

When the first packet is returned from the controller through the *packet_out* message, the specified short destination address filed in its header will be bound with the $\lceil \log_2(N_C) \rceil$-bit unique signature. It forms a *signature hash table entry* and will be stored in the *signature hash table*. For example, if there are 8 TCAM chips in a switch, we need 3 bits to denote the index of each TCAM chip. If a new packet with IP destination address 10.0.0.2 sent to the controller triggers the installation of a forwarding rule, the controller may create a 3-bit signature, say 001, to specify the target installation TCAM chip of the associated forwarding rule. When this packet is received by the ingress switch through an OpenFlow *packet_out* message, the signature 001 can be parsed, and then a *signature hash table entry* ⟨dest_addr:10.0.0.2, chip_index:001⟩ will be created together with the destination address and stored in the *signature hash table*. Once the successive packets belonging to the same session arrive at the hash table, they will be directed to the target TCAM chip with ID 001. Note that, such a signature hash table can be realized by an ASIC-based comparator [3] or a very small TCAM chip such as the pre-classifier proposed in [5].

After the *lookup localization*, the packet is directed to a designated TCAM chip and compared to the stored rules. Furthermore, the hash table entries can also be updated by the controller flexibly and dynamically when the assigned target TCAM chip of a session is changed according to a certain traffic engineering policy.

If a TCAM chip is being installed with rules, we call it an *activated* chip. All activated TCAM chips work under the scheduled voltage/frequency settings. If a match is hit, the corresponding action will be executed to the target packet, which is finally delivered out of the switch. Otherwise, the packet may be dropped or sent to a controller for further consultation.

Open Issues

Some open issues in our proposed Green DataPath architecture are summarized as follows.

Implementation of DVFS-Enabled Switch: In order to support DVFS working mode, new hardware modules, that is, the *signature hash table* and *voltage/frequency tuning* modules, are expected to be embedded into commercial SDN switches as the next generation of switches.

New Protocol Design: Corresponding to the proposed architecture, new supporting protocols or extensions of existing SDN protocols aim to enable SDN networks to work under energy-efficient DVFS control. For instance, the control plane requires new application programming interfaces (APIs) that can collect the DVFS parameters from switches and distribute the individual voltage/frequency configurations to each switch.

Fine-Grained Traffic Engineering: It is also desired to perform energy-aware fine-grained traffic engineering by jointly considering the DVFS parameter settings under our proposed architecture, aiming to achieve low latency, high throughput, and low power consumption.

Green DataPath Finding Problem

This section focuses on a green datapath finding problem under the proposed architecture. First, we describe this problem, and then propose a heuristic algorithm to find the energy-efficient data path for each target traffic session in the context of traffic engineering.

PROBLEM STATEMENT

We consider an SDN network $\mathcal{G} = (N, E)$, which consists of controllers (one or more), DVFS-enabled switch set N, and edge set E. Without loss of generality, as shown in Fig. 2b, we assume that all the switches are homogeneous, and each of them is equipped with N_c TCAM chips. Given the date rate requirement of a set U of traffic sessions, as well as the reliable voltage/frequency tuning ranges, a controller calculates a routing path for each of them, and decides on which TCAM chip to install the corresponding rules and the voltage/frequency configuration parameters of the active TCAM chips. Based on the system model described above, we study a *Green data path finding problem* (GDP) with the objective of minimizing the total energy consumption of all traffic sessions such that the end-to-end traffic rate constraints and TCAM chip's rule space constraints are obeyed.

ALGORITHMS

In order to solve GDP, we propose a framework of a DVFS-based algorithm that can be executed on an SDN controller. The flowchart of such an algorithm framework is shown in Fig. 3. Since the target is to save energy over all sessions while TCAM-based flow tables are being used, we fix the operating voltage of TCAM chips at the lowest reliable value and tune the operating frequency of TCAM chips in the proposed algorithm.

At first, the controller finds a candidate path set P_i using the Depth-First-Search (DFS) algorithm for each session $i \in U$, and picks up the current feasible shortest one for each session subject to the bandwidth constraints on links. Then all the sessions are sorted in a specific order according to their demanded traffic rates (increasing or decreasing) or their original arrival sequence. The corresponding traffic scheduling schemes are denoted as first-fit-increasing (FFI), first-fit-decreasing (FFD), and first-in-first-serve (FIFS). Next, each session in the sorted set U' is going to be provisioned in sequence by checking each traversed switch in the selected shortest routing path. If there is no feasible TCAM chip in the current switch, the infeasibility notification will be returned. Otherwise, assign the required forwarding rules in the first feasible TCAM chip for the current target session. Note that we call a TCAM chip feasible if its available frequency and remaining flow table space are able to hold the current target session. Afterward, the controller calculates the lowest voltage/frequency configuration that incurs the lowest power consumption to the current TCAM chip subject to the traffic rate constraints. The algorithm finishes successfully once all the sessions have been processed.

PERFORMANCE EVALUATION

This section presents the simulation results of the performance evaluation under the proposed architecture. In simulation we adopt the CORONET [15] topology, which consists of 60 switch nodes and 79 bidirectional links. Using the CORONET CONUS topology, we enforce four data source servers connecting to four switch nodes that are located in Salt Lake City, Utah; Dallas, Texas; Louisville, Kentucky; and Scranton, Pennsylvania. Then, in total, 56 traffic sessions are generated between server-connected switch nodes and other switch nodes.

The traffic rate of each session is randomly generated within the range [12, 1200] Gb/s. We assume that packets are transmitted via TCP, in which the size of each packet is 1500 bytes. In that case, if the traffic rate of a session is 1200 Gb/s, the corresponding packet rate is calculated as (1200 Gb/s)/(1500 × 8 b/packet) = 100 million packets per second (Mpps). By invoking the DFS algorithm, we provide each session with five candidate paths. Furthermore, each session is assumed to consume only one forwarding rule in every traversed switch, and each switch is assumed to be assembled with eight TCAM chips. For each chip, we fix the reliable operating voltage as 1.5 V referring to the parameter settings shown in [13], and vary the maximum tunable operating frequency (MOP-frequency) within a reliable range. In our simulation study, four performance metrics are considered: throughput, the number of activated TCAM chips, power consumption, as well as the feasible solution ratio (FSR), which is defined as the feasible sessions over all sessions.

In the first group of simulations, we aim to compare the FSR and throughput performance among three traffic scheduling schemes (i.e., FFI, FFD, and FIFS). The MOP-frequency of each TCAM

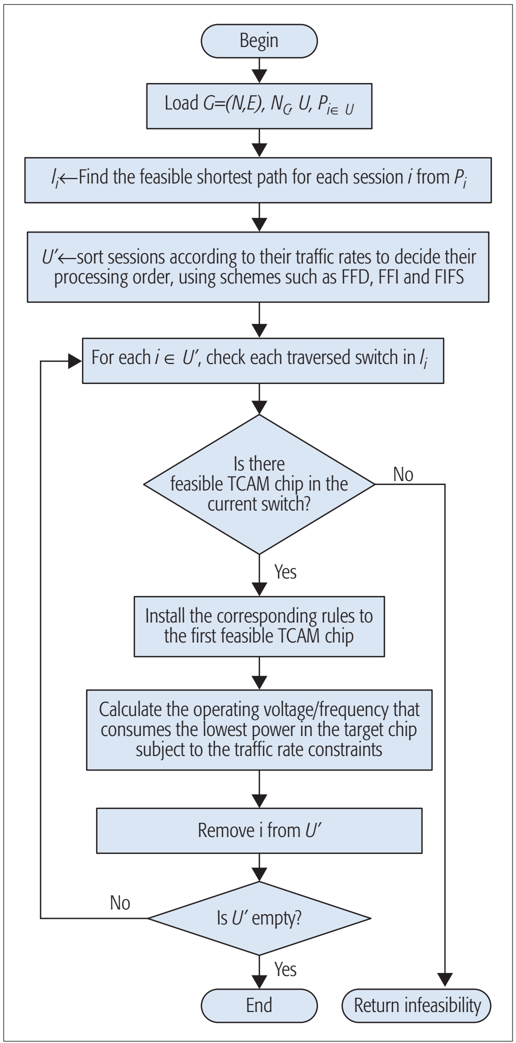


Figure 3. The DVFS-based algorithm framework.

Using the CORONET CONUS topology, we enforce four data-source servers connecting to four switch nodes that are located in Salt Lake City, Dallas, Louisville, and Scranton. Then, in total, 56 traffic sessions are generated between server-connected switch nodes and other switch nodes.

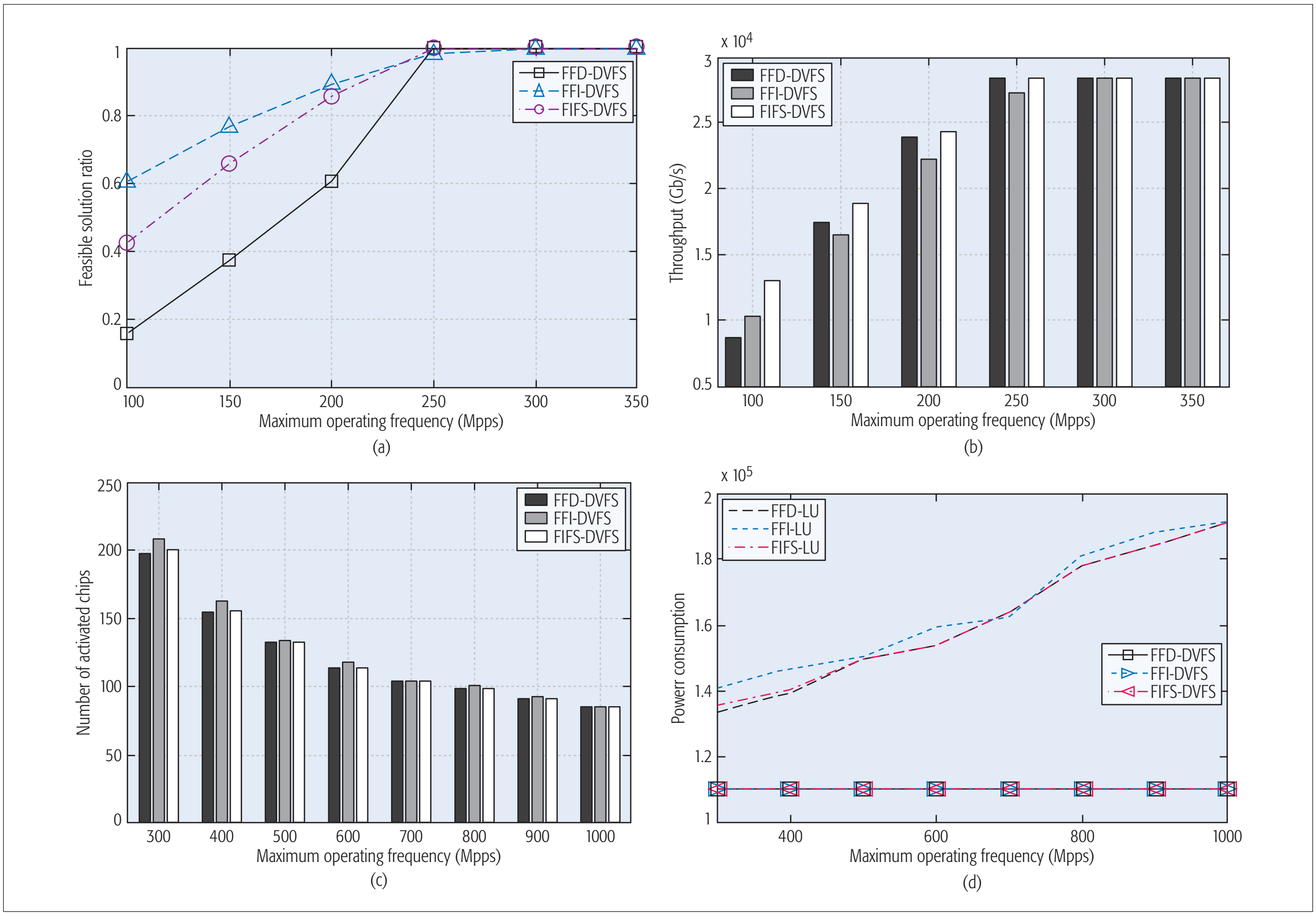


Figure 4. Performance evaluation while tuning the maximum tunable operating frequency of TCAM chips: a) feasible solution ratio; b) throughput; c) total number of activated TCAM chips; d) power consumption.

chip varies within [100, 350] Mpps or MHz. As demonstrated in Fig. 4a, the FFI always has the highest capability to find solutions for sessions, while FFD is shown to be the worst. The reason is that when the maximum operating frequency of a TCAM chip is too low, say less than 300 Mpps, it is infeasible to find a solution for the unassigned sessions once there is no available TCAM chip. In the FFD scheme, the session with the larger traffic rate has higher priority to be processed by the proposed DVFS-based algorithm. As a result, most critical TCAM frequency is consumed by a small number of large-sized sessions. Under the FFI scheme, the situation is exactly opposite. The performance of the FIFS scheme is in between.

Figure 4b shows the total throughput of the three DVFS-based schemes. It can be observed that the FIFS always has the highest throughput, and its advancement is significant especially when the MOP-frequency is less than 250 Mpps. The throughput of FFD is lower than FFI only when MOP-frequency is 100 Mpps, but higher afterward. Although we already know that FFI has a higher FSR than FFD from Fig. 4a, the accumulated traffic rate over all satisfied sessions would be lower than FFD, due to the fact that the giant-sized sessions can be satisfied with higher priority under the FFD scheme. Finally, as the MOP-frequency grows high enough, all the schemes achieve the same throughput because the feasible solution of all sessions can always be found.

Next, we increase the MOP-frequency of TCAM chips to 1000 Mpps, aiming to evaluate the number of activated chips and power consumption when all sessions can be satisfied with feasible routing solutions. We can observe from Fig. 4c that the total number of activated TCAM chips shows as a decreasing function of MOP-frequency. Although all schemes perform similarly, FFI activates slightly more chips than the other two schemes. We attribute this to the fact that the small-sized sessions are always provisioned first in each TCAM chip, and there will be more waste of frequency resource in the activated chips. As a result, more chips need to be activated to satisfy the giant-sized sessions. However, when the MOP-frequency becomes very high, the differences among schemes disappear.

Finally, to evaluate the power saving introduced by DVFS, we compare our proposed algorithms with a non-DVFS-based solution, denoted as the least unified (LU) strategy, while the MOP-frequency of a TCAM chip varies within [300, 1000] Mpps. In LU, the voltage/frequency setting in all TCAM chips is unified. Furthermore, the TCAM chip should be configured with the fewest operating settings that ensure all sessions are processed without queuing delay in switches. The power consumption under the two policies on TCAM chips are shown in Fig. 4d, based on the power consumption model given above. We can observe that the proposed

DVFS-based algorithms save power consumption by 20–40 percent, and the power consumption of the LU strategy increases with MOP-frequency in an approximately linear manner. The reason is that the total power consumption is only determined by the summation of traffic rates under the proposed algorithms. In contrast, the frequency capability of each activated TCAM chip will be almost fully exploited when the MOP-frequency increases from 300 Mpps to 1000 Mpps under the LU strategy. We also notice that the aggregated traffic rate in the switch nodes near servers is significantly higher than the averaged workload that is assigned to other switch nodes. In order to satisfy the traffic rate requirement in the heavy-loaded nodes, the *lowest* frequency must be tuned to a very high value, thus increasing the overall power consumption under the unified setting policy.

In summary, we have the following observations:
- Too low frequency settings may degrade the feasible solution ratio and the total throughput.
- The number of the activated TCAM chips decreases while increasing the chip's operating frequency.
- The power consumption does not grow even when the maximum tunable operating frequency is large enough under the proposed Green DataPath architecture.

Conclusion

In this article, a novel Green DataPath architecture has been proposed to find energy-efficient routing paths for traffic sessions. We focus on how to achieve energy-efficient traffic engineering in the TCAM-based SDN networks via introducing the DVFS power management technique. We have designed an DVFS-enabled Green DataPath architecture for SDN networks, where the dedicated SDN controlling a DVFS scheduling module and DVFS-enabled switches is presented. Then, for the proposed architecture, an energy-efficient data path finding algorithm framework has been proposed. Finally, we have evaluated this framework under three traffic scheduling schemes (i.e., FFD, FFI and FIFS) by extensive simulations. Some useful insights of parameter settings have been discussed and summarized.

Acknowledgment

This work was partially supported by Grant-in-Aid for Japan Society for the Promotion of Science (JSPS) Fellows, Grant Number 16J07062.

References

[1] H. Yamanaka *et al*., "Openflow Networks with Limited l2 Functionality," *Proc. 13th Int'l. Conf. Networks*, 2014, pp. 221–29.

[2] P. T. Congdon *et al*., "Simultaneously Reducing Latency and Power Consumption in Openflow Switches," *IEEE/ACM Trans. Networking*, vol. 22, no. 3, 2014, pp. 1007–20.

[3] R. Panigrahy and S. Sharma, "Reducing Tcam Power Consumption and Increasing Throughput," *Proc. 10th Symp. High Performance Interconnects*, 2002, pp. 107–12.

[4] F. Zane, G. Narlikar, and A. Basu, "Coolcams: Power-Efficient tcams for Forwarding Engines," *Proc. IEEE 22nd Annual Joint Conf. IEEE Computer and Commun.*, vol. 1, 2003, pp. 42–52.

[5] Y. Ma And S. Banerjee, "A Smart Pre-Classifier to Reduce Power Consumption of Tcams for Multi-Dimensional Packet Classification," *Proc. ACM SIGCOMM 2012 Conf. Applications, Technologies, Architectures, and Protocols for Computer Commun.*, 2012, pp. 335–46.

[6] V. Ravikumar, R. N. Mahapatra, and L. N. Bhuyan, "Easecam: An Energy and Storage Efficient Tcam-Based Router Architecture for IP Lookup," *IEEE Trans. Comp*., vol. 54, no. 5, 2005, pp. 521–33.

[7] C. R. Meiners, A. X. Liu, and E. Torng, "Tcam Razor: A Systematic Approach Towards Minimizing Packet Classifiers in Tcams," *Proc. IEEE Int'l. Conf. Network Protocols*, 2007, pp. 266–75.

[8] Y. Sun and M. S. Kim, "Tree-Based Minimization of Tcam Entries for Packet Classification," *Proc. 7th IEEE Consumer Commun. and Networking Conf.*, 2010, pp. 1–5.

[9] K. Kannan and S. Banerjee, "Compact Tcam: Flow Entry Compaction in Tcam for Power Aware SDN," *Proc. Distrib. Comp. and Networking*, Springer, 2013, pp. 439–44.

[10] H. Huang *et al*., "Joint Optimization of Rule Placement and Traffic Engineering for Qos Provisioning in Software Defined Network," *IEEE Trans. Computers*, vol. 64, no. 12, 2015, pp. 3488–99.

[11] N. McKeown *et al*., "Openflow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 38, no. 2, 2008, pp. 69–74.

[12] K. Pagiamtzis and A. Sheikholeslami, "Content-Addressable Memory (Cam) Circuits And Architectures: A Tutorial And Survey," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, 2006, pp. 712–27.

[13] S. Borkar *et al.*, "Parameter Variations and Impact on Crcuits and Microarchitecture," *Proc. 40th Annual Design Automation Conf.*, 2003, pp. 338–42.

[14] D. C. Snowdon, S. Ruocco, and G. Heiser, "Power Management and Dynamic Voltage Scaling: Myths and Facts," *Proc. 2005 Wksp. Power Aware Real-Time Computing*, vol. 12, 2005.

[15] A. L. Chiu *et al*., "Architectures and Protocols for Capacity Efficient, Highly Dynamic and Highly Resilient Core Networks," *IEEE/OSA J. Optical Commun. Net*., vol. 4, no. 1, 2012, pp. 1–14.

Biographies

HUAWEI HUANG received both his Bachelor's and Master's degrees in computer science and technology from the China University of Geoscience, Wuhan, in 2011 and 2013, respectively. He is currently a Ph.D. student at the University of Aizu, Japan. His research interests are in software-defined networking, network functions virtualization, and wireless networks. He is now a research fellow of the Japan Society for the Promotion of Science.

SONG GUO [M'02, SM'11] received his Ph.D. degree in computer science from the University of Ottawa, Canada. He is a full professor at the School of Computer Science and Engineering, University of Aizu. His research interests are mainly in the areas of wireless communication and mobile computing, cloud computing and networking, and cyber-physical systems. He serves as Associate Editor of *IEEE TPDS* and *IEEE TETC*. He is a Senior Member of ACM.

JINSONG WU is an associate professor in the Department of Electrical Engineering at the Universidad de Chile, Santiago. He is the founding Chair of the IEEE Technical Committee on Green Communications and Computing. He is an Editor of the *IEEE JSAC* Series on Green Communications and Networking. He was the leading editor and co-author of the comprehensive book *Green Communications: Theoretical Fundamentals, Algorithms, and Applications* (CRC Press, 2012).

JIE LI [M'94, SM'04] received his Dr. Eng. degree from the University of Electro-Communications, Tokyo, Japan. He is currently a professor at the University of Tsukuba, Japan. His current research interests include mobile distributed computing and networking, big data and cloud computing, the Internet-of Things, operation systems, and modeling and performance evaluation of information systems. He is a Senior Member of the ACM and a member of the Information Processing Society of Japan.

We also notice that, the aggregated traffic rate in the switch nodes near servers is significantly higher than the averaged workload that is assigned to other switch nodes. In order to satisfy the traffic rate requirement in the heavy-loaded nodes, the least frequency must be tuned to a very high value, thus increasing the overall power consumption under the unified setting policy.