An Analytical Model for Combined SDN Forwarding Element

Qinglei Qi, Wendong Wang, Xiangyang Gong, Xirong Que
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications, Beijing 100876, China
{qql, wdwang, xygong, rongqx}@bupt.edu.cn

***Abstract*—Recent studies have shown that the flow table size of hardware SDN switch cannot match the number of concurrent flows. Combined SDN Forwarding Element (CFE), which comprises software switch and hardware switch, becomes an alternative approach for tackling this problem. Because software switch has lower lookup speed than hardware switch, different proportions of traffic allocated to software switches in CFE have different effects on the delay bounds of all flows entering CFE. As delay-guarantee is a nontrivial task for network providers, especially with the increasing number of delay-sensitive applications, a model to analyze the delay bound given a flow allocation in CFE is important. With the one-to-one correspondence between flow allocation and rules placement solution, the analytical model can be used to evaluate and compare rules placement solutions and provide a basis for designing better rules placement solution in CFE. In this paper, we propose an analytical model for CFE based on network calculus, and then validate this model through simulations in NS-3.**

I. Introduction

Since control functions of networking devices are centralized onto the controller, software defined networking enables control plane to be programmable. Therefore, SDN will play an important role in 5G network to implement network function virtualization (NFV) and to reduce capital and operations cost. OpenFlow [1] is the most popular protocol in SDN. In OpenFlow networks, forwarding rules are generated by the controller, and are distributed into the flow tables of switches.

The size and lookup speed of flow tables are the key metrics of switch performance. Openflow switches can be classified into software switches and hardware switches [2] based on the types of flow tables. Ternary Content Addressable Memory (TCAM) has become an indispensable choice of hardware switch because of its faster lookup speed. Nevertheless, the size of TCAM in hardware switch is limited as a result of the high cost and energy consumption of TCAM [3]. Recent studies have established that the flow table size of hardware switch can not match the number of concurrent flows [4] [5]. The gap will be enlarged in 5G era with the increasing number of flows. The flow table (e.g. SRAM) in software switch has conversely characteristics with TCAM, i.e., characteristics of slower lookup speed, lower cost and energy consumption. Combined SDN Forwarding Element (CFE), which comprises software switch and hardware switch, becomes a trade-off between the size and lookup speed of flow table.

Delay guarantee is of important significance to network providers with the increasing of delay-sensitive application. The end-to-end delay guarantee problem can be decomposed into a set of single-hop delay guarantee problems along each data flow in the network [6] [7]. For satisfying the delay requirement of flow at a CFE, the model to evaluate the rules placement solution in terms of delay bound is indispensable. If the delay bounds of flows caused by a placement solution can be analyzed in advance, a desired rules placement solution (e.g., a solution that satisfy the delay requirements of maximum number of flows) can be obtained. As there is a one-to-one correspondence between flow allocation and rules placement solution, the delay bounds of flows corresponding to a specific rules placement solution in CFE can be evaluated indirectly by the model, which describes the relationship between the flow allocation in CFE and the delay bounds of flows.

Therefore, we propose an analytical model for CFE based on network calculus, and then validate this model through simulation in NS-3.

II. Related work

CacheFlow [8] is an ingenious implementation of CFE. The essence of CacheFlow is to take TCAM as a cache memory and to offload most rules, which means exact-match rules except where noted in this paper, into software switch. The rule-caching algorithms based on the rule dependencies and traffic counts in CacheFlow attained higher cache-hit rate. However, an analytical model for CFE is still necessary. From a statistical standpoint, the packets in the flows with low rate or small size have higher probability of being redirected to software switch than the other packets in CacheFlow. Thus, CacheFlow will likely cause higher delay bound for the flows with low rate. In fact, the flows with low rate may be more sensitive to delay than the flows with high rate [9]. Azodolmolky, S. et al. [10] utilized network calculus to analyze the behavior of an SDN switch and the interaction between SDN switch and controller. However, they did not consider the case that multiple switches are combined together.

Queueing theory [11] and network calculus [12] are two important analytical methods for communication network. The former emphasizes on the quantities in an equilibrium, while the latter focuses on the performance guarantees. In addition,

978-3-901882-89-0 @2017 IFIP

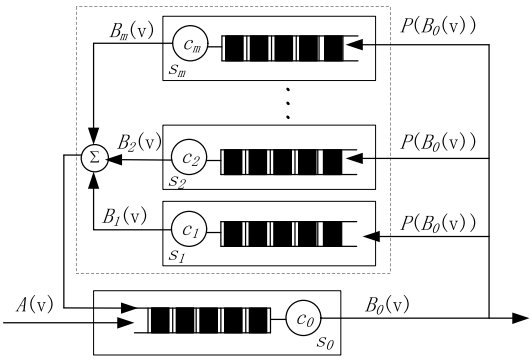


Fig. 1: Network calculus model for CFE

the self-similarity of internet traffic makes the queueing theory difficult to analyze the performance of computer network [13]. Florin ciucu [14] demonstrates that network calculus can capture the behavior of computer network tightly. Therefore, we adopt network calculus in the analytical model.

III. Delay analysis of CFE

A. Network calculus basics

The reference [12] presents the definitions of arrival curve and service curve, and prove five basic theorems (i.e., Delay Bound, Output Characterization, Concatenation Property, Leftover Service and Superposition). In addition, Routing theorem for an ideal router and the definition and theorem for nonfeedforward routing are provided in [12] as follows.

Theorem 1 (Routing): For an ideal router, if A is (σ, ρ)-upper constrained and P is (δ, γ)-upper constrained, then B is $(\gamma\sigma + \delta, \gamma\rho)$-upper constrained.

Definition 1: For any increasing sequence A, define its stopped sequence at time τ, denoted by A^τ, by

$$A^\tau(v) = \begin{cases} A(v) & \text{if } v \leq \tau, \\ A^\tau & \text{otherwise.} \end{cases}$$

Theorem 2: For every ρ, a stopped sequence A^τ is $(\sigma(\tau), \rho)$-upper constrained, where

$$\sigma(\tau) = \max_{0 \leq v \leq \tau} \max_{0 \leq u \leq v} [A(v) - A(u) - \rho(v-u)].$$

Corollary 1: If $A^{(\tau)}$ is (σ, ρ)-upper constrained, then $\sigma(\tau) \leq \sigma$, where $\sigma(\tau)$ is defined in Theorem 2.

B. Network calculus model of CFE

The relationship between the hardware switch and the software switches in CacheFlow [8] are exploited in CFE architecture. However, the coordinator of switches in CFE can be also controller expect a local arbitrator similar with Cache Master in CacheFlow. The local arbitrator can adjust the rules distribution in CFE more efficient than the controller. However, the horizon of the local arbitrator is limited. The trade-off between the local arbitrator and the controller is a separate research project. The comprehensive solutions to address this issue will be investigated in our future work.

As illustrated in Fig.1, the network calculus model of CFE comprises multiple switches, each of which is considered as a work conserving server with an infinite queue. The capacity of the switch s_i is c_i $(0 \leq i \leq m)$. s_0 denotes the hardware switch, while s_j denotes the j^{th} software switch $(1 \leq j \leq m)$. For brevity, we divide CFE into multiple channels based on the switches comprised. The channel $s_0 \rightarrow s_j \rightarrow s_0$ is called channel j, while the channel that includes only s_0 is called channel 0 for uniformity. The channel 0 is also called fast channel, while the other channels are all called slow channel.

Let $A_i \sim (\sigma_i, \rho_i)$ denote the arrival process of the aggregated flows that are allocated to channel i. Because the numbers of packets, which are forwarded to every software switch, are almost equal, $A_1(v) ... = A_j(v) ... = A_m(v)$. Then the arrival process of the aggregated flow that arrives hardware switch from outside of CFE is $\sum_{l=0}^{m} A_l \sim (\sum_{l=0}^{m} \sigma_l, \sum_{l=0}^{m} \rho_l)$, which is denoted as $A \sim (\sigma, \rho)$. Furthermore, let $P \sim (\delta, \gamma)$ denote the routing sequence which affects the rate of the flow from the hardware switch to each software switch and $B_i(v)$ denote the output from s_i. Considering that the capacities of all software switches are same, $B_1(v) ... = B_j(v) ... = B_m(v)$.

C. Delay bound analysis for CFE

Let $\tilde{A}_i$ denote the overall arrival process to the s_i. Thus, we have

$$\tilde{A}_0(v) = A(v) + mB_j(v), \tag{1}$$

$$\tilde{A}_j(v) = P(B_0(v)). \tag{2}$$

Let B_i^τ be the stopped sequences of s_i at time τ. From Theorem 2, for any a_i, B_i^τ is $(\sigma_i(\tau), a_i)$-upper constrained, where $\sigma_i(\tau) = \max\limits_{0 \leq v \leq \tau} \max\limits_{0 \leq u \leq v} [B_i(v) - B_i(u) - a_i(v-u)]$.

Now we choose a_0 and a_j to be the solution of the following equations

$$a_0 = \rho + ma_j, \tag{3}$$

$$a_j = \gamma a_0. \tag{4}$$

Through solving (3) and (4), under the case $\gamma < 1/m$, one can obtain

$$a_0 = \rho/(1 - m\gamma), \tag{5}$$

$$a_j = \gamma\rho/(1 - m\gamma). \tag{6}$$

Based on the Output Characterization Theorem, the average rate of a flow leaving a work-conserving switch is equal to the rate of that flow entering the switch. Thus, we have

$$a_0 = 2\rho - \rho_0. \tag{7}$$

Through solving (5) and (7), under the case $\gamma < 1/m$, one can obtain

$$\gamma = (\rho - \rho_0)/(m(2\rho - \rho_0)). \tag{8}$$

Applying the Superposition Theorem to the equation (1) $\tilde{A}_0(v) \sim (\sigma + m\sigma_j(\tau), \rho + ma_j)$. According to Theorem 1, Theorem 2 and (2), we have $\tilde{A}_j(v) \sim (\gamma\sigma_0(\tau) + \delta, \gamma a_0)$. From the Output Characterization theorem, we can obtain $B_0^\tau(v) \sim (\sigma + m\sigma_j(\tau), \rho + ma_j)$, and $B_j^\tau(v) \sim (\gamma\sigma_0(\tau) + \delta, \gamma a_0)$. It then follows from Corollary 1 that

$$\sigma_0(\tau) \leq \sigma + m\sigma_j(\tau), \tag{9}$$

$$\sigma_j(\tau) \leq \gamma\sigma_0(\tau) + \delta. \quad (10)$$

Solving (9) and (10) results in $\sigma_0(\tau) \leq \tilde{\sigma}_0$ and $\sigma_j(\tau) \leq \tilde{\sigma}_j$, where

$$\tilde{\sigma}_0 = (1 - m\gamma)^{-1}(\sigma + m\delta), \quad (11)$$

$$\tilde{\sigma}_j = (1 - m\gamma)^{-1}(\gamma\sigma + \delta). \quad (12)$$

As these bounds are independent of τ, (11) and (12) also hold for the unstopped sequences B_0 and B_j. Thus, we obtain

$$B_0 \sim (\tilde{\sigma}_0, a_0), \quad (13)$$

$$B_j \sim (\tilde{\sigma}_j, a_j). \quad (14)$$

These in turn imply that

$$\tilde{A}_0 \sim (\tilde{\sigma}_0, a_0), \quad (15)$$

$$\tilde{A}_j \sim ((\tilde{\sigma}_j, a_j). \quad (16)$$

Because only the flows allocated to the channel j will arrive the switch j, it holds that

$$(\sigma - \sigma_j)/m \leq \tilde{\sigma}_j \leq \sigma/m. \quad (17)$$

In conjunction with (8), (12) and (17), we get $(\rho_0\sigma - \rho\sigma_0)/(m(2\rho - \rho_0)) \leq \delta \leq (\rho_0\sigma)/(m(2\rho - \rho_0))$.

Let g_0 denote the service curve of the switch s_0 for A_0, according to the Leftover Service theorem, we have

$$g_0(v) = (c_0 v - (\tilde{A}_0(v) - A_0(v)))^+, \quad (18)$$

where $(w)^+ \triangleq \max\{0, w\}$.

Applying the Superposition Theorem and the conclusion of (15) to (18) yields $g_0(v) = ((c_0 - a_0 + \rho_0)t - \tilde{\sigma}_0 + \sigma_0)^+$.

Let g'_j denote the service curve of the switch s_0 to the arrival process of the flows through channel j. According to the Leftover Service theorem, we have

$$g'_j(t) = (c_0 v - (\tilde{A}_0(v) - A_j))^+. \quad (19)$$

Applying the Superposition Theorem and the conclusion of (15) to (19) yields $g'_j(v) = ((c_0 - a_0 + \rho_j)v - \tilde{\sigma}_0 + \sigma_j)^+$.

Let g''_j denote the service of s_j to the $P(B_0(v))$. According to Theorem 3, we have $g''_j(v) = c_j v$.

Let g'''_j denote the service curve of the switch s_0 to B_j, according to the Leftover Service theorem, we have

$$g'''_j(v) = (c_0 v - (\tilde{A}_0(v) - B_j(v))^+. \quad (20)$$

Applying the Superposition Theorem and the conclusion of (14) and (15) to (20) yields $g'''_j(v) = ((c_0 - a_0 + a_j)v - \tilde{\sigma}_0 + \tilde{\sigma}_j)^+$.

Let g_j denote the service curve of CFE to the external arrival process of the flows through channel j. According to the Concatenation Property Theorem, we have $g_j(v) = g'_j \otimes g''_j \otimes g'''_j(v)$.

Let $\alpha_i(v) = \rho_i v + \sigma_i$ be the arrival curve of the aggregated flow that pass through channel i. According to the Delay Bound theorem, the delay $d_i(v)$ of the aggregated flow at time v is bounded by

$$d_i(v) \leq inf\{\tau \geq 0 : \alpha_i(u) \leq g_i(u + \tau), 0 \leq u \leq v\}. \quad (21)$$

TABLE I: average rates and burst sizes of token buckets constraining f_0, f_1, and f_2

	f_0		f_1		f_2	
	ρ_0(pps)	σ_0(pkts)	ρ_1(pps)	σ_1 (pkts)	ρ_2(pps)	σ_2(pkts)
case 1	20000	10000	5000	2500	0	0
case 2	17500	8750	7500	3750	0	0
case 3	20000	10000	2500	1250	2500	1250
case 4	17500	8750	3750	1875	3750	1875

IV. Model validation and performance evaluation

In this section, we compare the experimental value and the theoretical value of the maximum delay of the aggregated flow allocated to each channel of CFE under multiple cases. The experiment is conducted in NS-3. The theoretical value is calculated based on (21).

A. Simulation Setup

CFE is constituted by three openflow switches. The capacity of one switch, which acts as the hardware switch, is 0.2 million packets per second (Mpps). The capacities of the other two switches, which acts as the software switches, are both 0.01Mpps. There are seven concurrent flows passing through CFE. The sending host of each flow uses a token bucket filter (TBF) to constrain the flow. The size of the packet buffer is set larger than the bucket size of the same TBF. These flows are generated in terms of ON-OFF model. The *ON* time is a random number with an exponential distribution, while the *OFF* time is 0.5 seconds for the chance that token buckets are filled up.

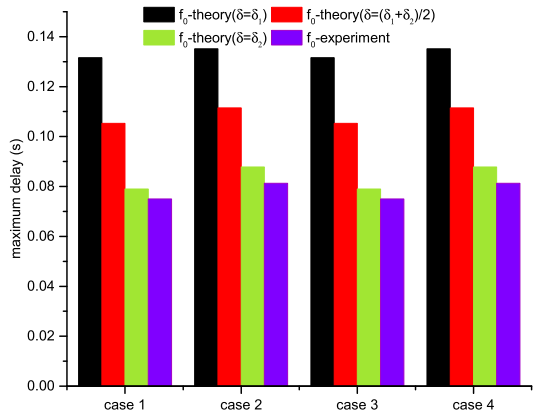
We take the time between entering and leaving CFE as the delay that a packet pass through CFE. To avoid the delay increased by link, the bandwidths and delays of all links are 100Gpps and 0ms. To prevent understating the real delay because of packet loss, the queues of all ports of all switches can contain 0.4M packets, which is more than the sum of burst size of all token buckets.

Depending on rules placement solutions, there are four different flow allocations among channels of CFE. Only one slow channel or software switch is active in both case 1 and case 2, while two slow channels are active in case 3 and case 4. Table I presents the aggregated flow characters in these four different cases. As declared in section III, f_i denotes the aggregated flow passing through channel i in CFE, ρ_i and σ_i denote separately the average rate and burst size of token bucket constraining f_i. Thus, as definition in section III, $\rho = \sum_{k=0}^{m} \rho_k$ and $\sigma = \sum_{k=0}^{m} \sigma_k$, where $m = 1$ in case 1 and case 2 while $m = 2$ in case 3 and case 4.

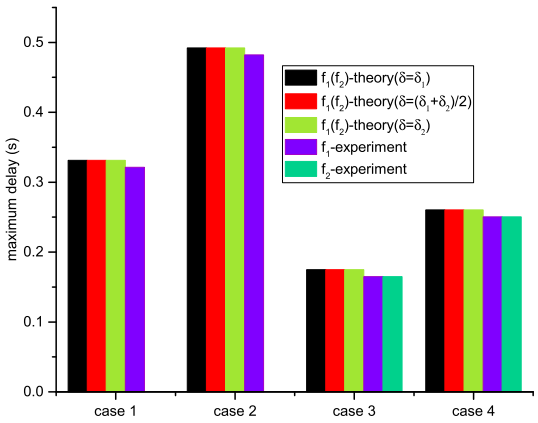
B. Results

Fig.2 depicts the experimental value of the maximum delay of any aggregated flow under four cases, and their three theoretical values. Three theoretical values of each aggregated flow under every cases are obtained respectively when $\delta = \delta_1 = (\rho_0\sigma)/(m(2\rho - \rho_0))$, $\delta = \delta_2 = (\rho_0\sigma - \rho\sigma_0)/(m(2\rho - \rho_0))$, and $\delta = (\delta_1 + \delta_2)/2$.

Four results can be observed through the comparisons among the results shown in Fig.2. First, through the comparisons between the results under case 1 and case 2 and between



(a) f_0



(b) f_1 and f_2

Fig. 2: Experimental values and theoretical values of the maximum delays of f_0, f_1 and f_2

the results under case 3 and case 4, it can be found that the maximum delay of any aggregated flow increases with the sum of ρ_1 and ρ_2 and the sum of σ_1 and σ_2. Thus, where the rule of a flow should be placed in CFE not only depends on the delay requirement of this flow, but also on the effect what this flow brings to the delays of other flows.

Second, through the comparisons between the results under case 1 and case 3 and between the results under case 2 and case 4, it can be found that the maximum delays of the aggregated flows, which are allocated to slow channels, decrease with the increasing number of the software switch.

Third, the theoretical value of the maximum delay of f_0 is more precise approximation with the experimental value when $\delta = \delta_2$.

Last, the theoretical values of the maximum delays of f_1 and f_2 are always tight approximation with the experimental value, and do not change with δ. The reason is that s_1 and s_2 are the bottlenecks of service providers to f_1 and f_2 respectively. So the services got by f_1 and f_2 in CFE are independent with δ.

V. Conclusion and Future Work

TCAM is a key enabler of hardware SDN switch to implementing line-speed forwarding. However, the high cost-to-density ratio and power consumption of TCAM limit the flow table size of hardware switch. The combination of software switch and hardware switch becomes an alternative approach for obtaining larger flow table. Given there is a lack of consideration on delay of rules placement in CFE, we propose an analytical model for CFE based on the network calculus and validate this model through simulation in NS-3. This analytical model can be used to predict the worst-case delay of each flow for a rules placement solution. In other words, the analytical model can provide a basis for rules placement with delay guarantee in CFE. As manually placing rules given many concurrent flows and delay constraints can be a grind, built on this analytical model, we can further design efficient algorithm to automate the rules placement in CFE. We leave the design of efficient rules placement algorithm as future work.

Acknowledgment

This work was supported in part by National High-Tech Research and Development Program (863 Program) of China under Grant No.2015AA016101.

References

[1] "Openflow." [Online]. Available: http://archive.openflow.org/

[2] M. Kuźniar, P. Perešíni, and D. Kostić, "What you need to know about sdn flow tables," in *Passive and Active Measurement: 16th International Conference, PAM 2015, New York, NY, USA, March 19-20*, 2015, pp. 347–359.

[3] "Tcams and openflow: What every sdn practitioner must know." [Online]. Available: https://www.sdxcentral.com/articles/contributed/sdn-openflow-tcam-need-to-know/2012/07/

[4] S. Banerjee and K. Kannan, "Tag-in-tag: Efficient flow table management in sdn switches," in *10th International Conference on Network and Service Management (CNSM) and Workshop*. IEEE, 2014, pp. 109–117.

[5] X.-N. Nguyen, D. Saucez, C. Barakat, and T. Turletti, "Rules placement problem in openflow networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1273–1286, 2015.

[6] X. Cao, Y. Dong, and D. H.-C. Du, "Synchronized multi-hop scheduling for real-time traffic on sdns," in *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2015, pp. 1–8.

[7] X. Wang, X. Wang, L. Liu, and G. Xing, "Dutycon: a dynamic duty-cycle control approach to end-to-end delay guarantees in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 4, p. 42, 2013.

[8] N. Katta, O. Alipourfard, J. Rexford, and D. Walker, "Cacheflow: Dependency-aware rule-caching for software-defined networks," in *Proceedings of the Symposium on SDN Research*. New York, NY, USA: ACM, 2016, pp. 6:1–6:12.

[9] D. Carra, "Controlling the delay of small flows in datacenters," in *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, June 2014, pp. 70–75.

[10] S. Azodolmolky, R. Nejabati, M. Pazouki, P. Wieder, R. Yahyapour, and D. Simeonidou, "An analytical model for software defined networking: A network calculus-based approach," in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec 2013, pp. 1397–1402.

[11] N. Mehravari, *Queueing Theory*. John Wiley and Sons, Inc., 2001.

[12] J.-Y. L. Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*. Springer-Verlag, 2001.

[13] Y. Jiang, "Network calculus and queueing theory: Two sides of one coin: Invited paper," in *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS '09, ICST, Brussels, Belgium, Belgium, 2009, pp. 37:1–37:12.

[14] F. Ciucu and J. Schmitt, "Perspectives on network calculus: no free lunch, but still good value," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, 2012, pp. 311–322.