Rules Placement with Delay Guarantee in Combined SDN Forwarding Element

Qinglei Qi, Wendong Wang, Xiangyang Gong and Xirong Que
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876 - China
[e-mail: qql@bupt.edu.cn,wdwang@bupt.edu.cn,xygong@bupt.edu.cn,rongqx@bupt.edu.cn]
*Corresponding author: Wendong Wang

Received October 25, 2016; revised February 28, 2017; accepted April 5, 2017; published June 30, 2017

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 several software switches and a hardware switch, becomes an alternative approach to tackle this problem. Due to the limited capacity of software switch, the way to route concurrent flows in CFE can largely affect the maximum delay that a flow suffers at CFE. As delay-guarantee is a nontrivial task for network providers with the increasing number of delay-sensitive applications, we propose an analytical model of CFE to evaluate a rules placement solution first. Next, we formulate the problem of Rules Placement with delay guarantee in CFE (RPCFE), and present the genetic-based rules placement (GARP) algorithm to solve the RPCFE problem. Further, we validate the analytical model of CFE through simulations in NS-3 and compare the performance of GARP with three benchmark algorithms.

Keywords: Software Defined Networking (SDN), Rules Placement, Delay Guarantee

This work was supported in part by National High-Tech Research and Development Program (863 Program) of China under Grant No.2015AA016101 and No.2015AA015601.

https://doi.org/10.3837/tiis.2017.06.004 **ISSN : 1976-7277**

1. Introduction

Software defined network (SDN) separates the control function and the data function of networking devices. The SDN controller decides what the rules are and where the rules are placed, while the SDN switches store the rules and carry out the actions corresponding to the rules. A packet will be forwarded to the controller if no match is found in switch, therefore the delay of this packet will be increased. On the other hand, the lookup performance of the switch will also affect the delay of a packet. So, the size and lookup speed of flow tables are the key metrics of switch performance.

OpenFlow [1] has become the most popular protocol in SDN. Openflow switches can be classified into software switches and hardware switches [2, 3] 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 and its support for wildcard matching. Nevertheless, the size of TCAM in hardware switch is limited as a result of the high cost and energy consumption of TCAM [4]. Recent studies have established that the flow table size of hardware switch cannot match the number of concurrent flows [5,6,7]. The flow table (e.g. SRAM) in software switch, on the other hand, has slower lookup speed, lower cost and energy consumption. Combined SDN Forwarding Element (CFE), which comprises several software switches and a hardware switch, becomes a trade-off between the size and lookup speed of the flow table.

Most recently, [8] has proposed a CFE architecture named as CacheFlow. The essence of CacheFlow is to offload most rules (which means exact-match rules except where noted in this paper) into software switch and to take the hardware switch as a cache. The packets which match rules in hardware switch are forwarded immediately outside of CFE, while the other packets are forwarded to software switches. After being tagged in software switches, the second type of packets return to the hardware switch, and are forwarded outside of CFE based on the tag by the hardware switch. Thus, there are two different kinds of channels in CFE: fast channel and slow channel.

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 [9, 10]. Research on the delay guarantee in the CFE is a foundation to guarantee the end-to-end delay in SDN. The maximum delay that a flow can tolerant at CFE is called the CFE delay requirement for short. Whether the CFE delay requirements of the flows can be met depends on the rules placement in CFE.

In this paper, we propose an analytical model of CFE to evaluate a rules placement solution first. This model can estimate the maximum delays of the aggregate flows through the fast channel or the slow channel of CFE. Next, taking this model as a base, we formulate the problem of Rules Placement with delay guarantee in CFE (RPCFE), and present the genetic-based rules placement algorithm (GRPA) to solve the RPCFE problem. Further, we validate the analytical model of CFE through simulations in NS-3 and compare the performance of GRPA with three benchmark algorithms.

The rest of the paper is organized as follows. Section 2 describes the related work. In section 3, the CFE architecture and the delay analysis of CFE based on network calculus are elaborated. In section 4, the formulation of the RPCFE problem and the GRPA algorithm are

presented. Experimental results are represented in Section 5, followed by conclusions in Section 6.

2. Related Work

The related works on TCAM capacity problem can be divided into three categories. The first category tries to use less number of TCAM entries by compression scheme. Forwarding rules with the same action were compressed with the default rule [11]. The second category tries to evict inactive or less important rules to release TCAM entries for other rules. FlowMaster in [12] reuses the flow table by recording packet arrival interval of each flow and deleting the entries which are predicted to be stale. The third category tries to split the set of rules into different switches. With the support from software switches, CacheFlow [8] offload the memory burden for commodity switch with TCAM. A rule-caching algorithms for CacheFlow also be proposed to attained higher cache-hit rate according to the rule dependencies and traffic counts. Similar with CacheFlow, CFE architecture also belongs to the third category. Comparing with CacheFlow, however, we propose an analytical model of CFE to predict the delay bound of packet, which is necessary to guarantee the QoS requirement of flow.

Under the background of the contradiction between the low-cost requirement from network provider and the high-performance requirement from user, QoS provision in communication systems always be a challenge and attract a lot of research. Allocating source (queue, frequency, etc.) based on service requirement is common concept to address source constrain. In mobile cloud systems, a simulation model that handles traffic in queues heterogeneity network based on services priorities was developed [13]. For OBS (Optical Burst Switching) network, an approach was designed to increase the performance of cloud services provision by adjusting transmission modes according to given requirements [14]. To the best of our knowledge, however, no work considers how to place rules into CFE according to delay requirements of flows.

Moreover, monitoring system for delay [15, 16] were proposed for adjusting the manage policy. However, the delay from the feedback loop between network monitoring and network re-configuration will affect adversely QoS. Thus, it is necessary to propose the analytical model of CFE to predict the delay bound so that the rules can be placed in CFE reasonably.

Queuing theory [17] and network calculus [18] are two important analytical methods for communication network. The former emphasizes on the quantities in an equilibrium, whereas the latter focuses on the performance guarantees. In addition, the self-similarity of internet traffic makes the queuing theory difficult to analyze the performance of computer network [19]. Fortunately, network calculus can capture the behavior of computer network tightly [20]. Therefore, network calculus is utilized in our works to analysis the delay bound.

3. Performance analysis of CFE

3.1 CFE Architecture

As shown in **Fig. 1**, CFE comprises of one hardware switch and multiple software switches. Only the hardware switch connects with the forwarding elements outside of CFE. On the hardware switch, the port through which the forwarding element outside of CFE connects with the hardware switch is called external port, and the port through which the software switch inside of CEF connects with the hardware switch is called internal port. The forwarding rules

in the flow table of the hardware switch fall into three categories: the first category of rules are exact-match rules, which specifies the external port for the packets; the second category of rules are coarse-grained rules, which specifies the external port for the packets based on the specific field (e.g., vlan); the last category of rules are also coarse-grained rules, which specifies the internal port for the packets. The numbers of packets forwarded into all internal ports can be close to each other through configuring the last category of rules reasonably. The priorities of these three categories of rules decrease in sequence. The software switches have a copy of entire flow tables including all exact-match rules, which are used to modify the value of specific field (e.g., vlan) of the packets. There is a one-to-one correspondence between the value and the external port of the hardware switch. The value as a internal tag will be invalidated (e.g., through stripping vlan tag) before the packet is forwarded out from the external port of the hardware switch to the outside of CFE.

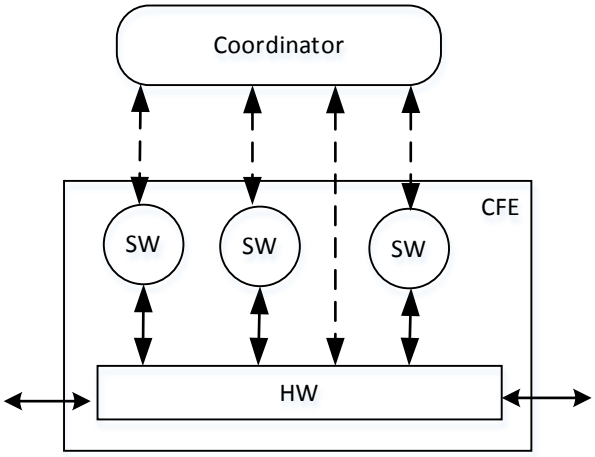


Fig. 1. CFE architecture.

The coordinator for switches in CFE can be either a local arbitrator (e.g., Cache Master in CacheFlow [8]) or controller. The local arbitrator can adjust the rules distribution in CFE more efficient than the controller. However, the horizon of the local arbitrator is limited. In delay guarantee case, the rules placement solution for CFE is generated based on the CFE delay requirements of flows. The CFE delay requirement of a flow depends on the maximum delay that the other forwarding elements on the same flow path cause. A local arbitrator is incompetence in this case, because the coordinator should know more than local information. The controller has opposite disadvantage and advantage with the local arbitrator. 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.

3.2 Network Calculus Overview

Network calculus is a filtering theory based on the min-plus algebra [21], and have been used for performance analysis of computer networks [22, 23]. The arrival curve and service curve are basic concepts of network calculus [24, 25].

Definition1(Arrival Curve): Let $\mathcal{F} = \{a(\cdot): \forall 0 \le x \le y, 0 \le a(x) \le a(y) \text{ and } \forall x \le 0, a(x) = 0\}$. A flow is said to have an arrive curve $\alpha \in \mathcal{F}$, if its arrival process $A(v)$ satisfies for all $0 \le u \le v$,

$$A(v)-A(u)\le\alpha(v-u).$$

Definition2(Service Curve): Consider a system S with input process $A(v)$ and output process $B(v)$. The system is said to provide to the input a service curve $\beta(v)\in\mathcal{F}$ ($\mathcal{F}$ has been defined in Definition 1) if for all $v\ge 0$,

$$B(v)\ge A\otimes\beta(v),$$

where, $a\otimes b(x)=\inf_{0\le y\le x}[a(y)+b(x-y)]$.

For arrival process, Cruz [22] defined the (σ,ρ) traffic characterization as follows.

Definition 3: A flow is said to be (σ,ρ) -upper constrained, denoted by $A\sim(\sigma,\rho)$, if for all $0\le u\le v$, there holds $A(v)-A(u)\le\rho(v-u)+\sigma$

Based on the concepts of the arrival curve for the traffic model and the service curve for the server model, Network Calculus has five basic Theorems (i.e., Delay Bound, Output Characterization, Concatenation Property, Leftover Service and Superposition) [24, 25]. For router and nonfeedforward routing, definition, theorems and corollary are given in [25] 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.

Definition4: For any increasing sequence A, its stopped sequence at time τ denoted by A^{τ}, where

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

Theorem2: For every ρ, a stopped sequence A^{τ} is $(\sigma(\tau),\rho)$ -upper constrained, where $\sigma(\tau)=\max_{0\le v\le\tau}\max_{0\le u\le v}[A(v)-A(u)-\rho(v-u)]$.

Corollary1: If $A^{(\tau)}$ is (σ,ρ) -upper constrained, then $\sigma(\tau)\le\sigma$, where $\sigma(\tau)$ is defined in Theorem 2.

3.3 Network calculus model of CFE

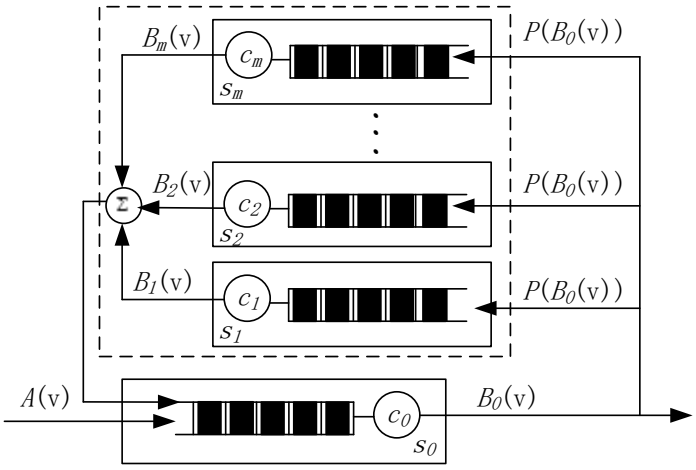


Fig. 2. Network calculus model of CFE.

As illustrated in **Fig. 2**, 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 $(0 \le i \le m)$. s_0 denotes the hardware switch, while s_j denotes the j^{th} software switch $(1 \le j \le m)$. For brevity, we divide CFE into multiple channels based on the switches comprised. The channel $s_0 \to s_j \to s_0$ 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 at hardware switch from outside of CFE is $\sum_{k=0}^{m} A_k \sim (\sum_{k=0}^{m} \sigma_k, \sum_{k=0}^{m} \rho_k)$, 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, $B_1(v)... = B_j(v)... = B_m(v)$.

3.4 Delay Bound Analysis for CFE

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

$$\tilde{A}_0(v) = A_0(v) + m(A_j(v) + B_j(v)), \tag{1}$$

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

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 $\sigma_i(\tau) = \max_{0 \le v \le \tau} \max_{0 \le u \le v} [B_i(v) - B_i(u) - a_i(v-u)]$.

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

$$a_0 = \rho + m a_j, \tag{3}$$

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

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) yields $\tilde{A}_0(v) \sim (\sigma + m\sigma_j(\tau), \rho + m a_j)$. 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 $B_0^{\tau}(v) \sim (\sigma + m\sigma_j(\tau), \rho + ma_j)$, $B_j^{\tau}(v) \sim (\gamma\sigma_0(\tau)+\delta, \gamma a_0)$.

It then follows from Corollary 1 that

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

$$\sigma_j(\tau) \le \gamma\sigma_0(\tau) + \delta. \tag{10}$$

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

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

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

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

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

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

These in turn imply that

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

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

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

$$\sigma_j \le \tilde{\sigma}_j \le \sigma / m. \tag{17}$$

In conjunction with (8) and (17), we get

$$(\rho_0\sigma - \rho\sigma_0)/(m(2\rho-\rho_0)) \le \delta \le (\rho_0\sigma)/(m(2\rho-\rho_0)).$$

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

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

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

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

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

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 the channel i. According to the Delay Bound theorem, the delay $d_{fast}(v)$ and $d_{slow}(v)$ of the aggregated flow through the fast channel and the slow channel at time v are bounded by

$$d_{fast}(v) \le inf\{\tau \ge 0 : \alpha_0(u) \le g_0(u+\tau), 0 \le u \le v\}, \quad (21)$$

$$d_{slow}(v) \le inf\{\tau \ge 0 : \alpha_j(u) \le g_j(u+\tau), 0 \le u \le v\}. \quad (22)$$

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 based on the analysis above.

4. Rules Placement with Delay Guarantee in CFE

4.1 Problem formulation

There are one hardware switch and m software switches in CFE. Let us define the flow table size of the hardware switch as T, the server rate of the hardware switch as c_0, and that of the software switch as c_1. There are a set of flows F, which pass concurrently through CFE. The weight of a flow $f \in F$ is ω_f. The flow f is constrained by a token bucket with the average rate ρ^f and burst size σ^f. The CFE delay requirement of the flow f is $\hat{l}_f$. The theoretical value of the delay bound at CFE for the flow f is l_f. The flow f will be allocated to the channel $x_f \in \{0,1...m\}$. We call $\sum_{f \in F'} \omega_f$ Delay Satisfaction Degree (DSD), where $F' = \{f \mid l_f \le \hat{l}_f \ and \ f \in F\}$. The objective of RPCFE is maximize of DSD and the objective function can be written as follows:

$$\max \sum_{f \in F'} \omega_f .$$

This objective function is optimized with satisfying the following conditions.

(a) The number of flows allocated fast channel is no larger than the flow table size of the hardware switch.

$$|\{f \mid x_f = 0\}| \le T .$$

(b) The sum of the average rates of the token buckets, which constrain the flows passing concurrently through the hardware switch, is no larger than the capacity of the hardware switch.

$$\rho_0 + 2\sum_{j=1}^{m} \rho_j \le c_0 .$$

(c) The sum of the average rates of the token buckets, which constrain the flows passing concurrently through the software switch, is no larger than the capacity of the software switch.

$$\rho_j \le c_1, \forall j \in \{1, 2...m\} .$$

(d) The weight of a flow is a random value in the range (0, 1).

$$0 < \omega_f < 1, \forall f \in F .$$

(e) The theoretical value of the delay bound at CFE for the flow f equals that for the aggregate flow through the channel to which the flow f is allocated.

$$l_f = \{d_j \mid x_f = j\}, \forall f \in F .$$

(f) The average rate corresponding to the aggregate flow through a channel equals the sum of the average rates of the token buckets, which constrain the flows through this channel.

$$\rho_i = \sum_{\{f \mid x_f = i\}} \rho^f, \forall i \in \{0, 1, 2...m\}$$

(g) The burst size corresponding to the aggregate flow through a channel equals the sum of the burst sizes of the token buckets, which constrain the flows through this channel.

$$\sigma_i = \sum_{\{f \mid x_f = i\}} \sigma^f, \forall i \in \{0, 1, 2...m\}$$

(h) A flow is allocated to the fast channel or one of m channels.

$$x_f \in \{0, 1...m\}, \forall f \in F .$$

It is a special case of RPCFE that the capacity of hardware switch is rather large while the capacities of software switches are very limited. Thus, the delay requirement of a flow is satisfied if and only if the rule of this flow is placed in the hardware switch in this special case. Obviously, this special case is equivalent to a 0-1 knapsack problem. It has been shown that 0-1 knapsack problem is a NP-hard problem [26]. Thus, delay-aware rules placement in CFE is a NP-hard problem.

4.2. Genetic-based rules placement algorithm

Genetic algorithm (GA) has been widely recognized to find the near-optimal solution to NP-hard problem. Thus, we present Genetic-based Rules Placement Algorithm (GRPA) for solving the RPCFE problem.

The pseudo-code of GRPA is shown as algorithm 1. GRPA generates *P* chromosomes (lines 1 in algorithm 1). Each chromosome presents one rules placement solution. A value of 0/1at the *i*th bit of a chromosome means that the rule of the *i*th flow is placed in the hardware/software switch. GRPA evaluates the chromosomes in population based on section 3 (lines 3-11 in algorithm 1). First, GRPA solves the arrive curves of the flows allocated the fast channel and the slow channel (line 5 in algorithm). Second, GRPA attains the service curves of CFE to the flows allocated the fast channel and the slow channel (line 6 in algorithm). Third, GRPA calculate the maximum delay of the aggregate flow passing through the fast channel and the slow channel of CFE (lines 7 in algorithm 1). Last, the fitness of the chromosome is solved based on the formula of the DSD (lines 8 in algorithm 1). GRPA exploits rank-based selection [27] to select *P* different chromosomes from the population to form the new population (line 24 in algorithm 1). Then, the population evolve through the crossover operator (line 25 in algorithm 1) and the mutation operator (line 26 in algorithm 1). The end

condition (line 12 in algorithm 1) is that the iterative number reach the max iterative number *IT* or that the optimal remains unchanged during CC iterations.

As shown in algorithm 2, the one-point crossover operator is adopted in GRPA. One pair of chromosomes is found based on *CP* first (lines 2-5 in in algorithm 2). Then the one-point crossover operator records all valid cross points of this pair of chromosomes (lines 6-13 in algorithm 2). Valid cross point refers to the point, to which the numbers of 0's are identical from the first point in the pair of chromosomes, is called valid cross point. The valid cross point can guarantee that the number of 0's keeps unchanged in a chromosome after crossover. One valid cross point is selected randomly as cross point (lines 14-15 in in algorithm 2). Finally, the one-point crossover operator adds two new chromosomes into the population (line 17 in algorithm 2). As shown in algorithm 3, the two-point mutation operator is adopted in GRPA. For each chromosome in population, the two-point mutation operator finds two random points first (lines 2-5 in in algorithm 2). If the values at these two points are different, they are swapped to generate a new chromosome (line 7 in in algorithm 2). In addition, the new chromosome is added the population (line 8 in in algorithm 2).

Algorithm 1: GRPA algorithm

Input: $F = \{< \rho^f, \sigma^f, l_f, \omega_f >\}, c_0, c_1, m$, T, initial population size P, crossover probability CP, max iterative number IT, convergence criteria CC

Output: The near optimal rules placement solution and its corresponding DSD.

1. $population \leftarrow initialPopulation(P, T, F)$
2. **while** $iterCounter \le IT$ **do**
3. **while** $i <$ sizeof$(population)$ **do**
4. $chromosome \leftarrow population[i]$
5. $(aC_{fast}, aC_{slow}) \leftarrow$ arriveCure$(F, m, chromosome)$
6. $(sv_{fast}, sv_{slow}) \leftarrow$ leftOverService$(F, m, c_0, c_1, aC_{fast}, aC_{slow})$
7. $(dB_{fast}, dB_{slow}) \leftarrow$ delayBound$(sv_{fast}, aC_{fast}, sv_{slow}, aC_{slow})$
8. $chromosome.fitness \leftarrow$ Evaluation$(F, dB_{fast}, dB_{slow}, chromosome)$
9. $population[i] \leftarrow chromosome$
10. $i++$
11. **end while**
12. **if** $iterCounter = IT$ **or** $isConvergent = 1$ **then**
13. **break**
14. **end if**
15. $record[iterCounter] \leftarrow$ optimal$(population, P)$
16. **if** $iterCounter > CC$ **then**
17. $chromosome1 \leftarrow record[iterCounter]$
18. $chromosome2 \leftarrow record[iterCounter - CC]$
19. **if** $chromosome1.fitness = chromosome2.fitness$ **then**
20. $isConvergent \leftarrow 1$

```
                continue
            end if
        end if
        population ← rankSelection(population, P)
        population ← crossOver(population, P, CP)
        population ← mutation(population, P)
        iterCounter++
    end while
    nearOptimalSolution ← optimal(population, P)
    return nearOptimalSolution
```

Algorithm 2: One-point crossover operator

Input: *population, P, CP*
Output: *population*

```
1.  while i < P − 1 do
2.      randomNum ← random(0,1)
3.      if randomNum < CP then
4.          father ← population[i]
5.          mother ← population[i+1]
6.          chromLen ← sizeof(father.character)
7.          while j < chromLen do
8.              if isValidCrossPoint(father, mother, j) then
9.                  validCrossoverPoint[validCrossPointNum] ← j
10.                 validCrossPointNum++
11.             end if
12.             j++
13.         end while
14.         crossPIndex ← random(1, validCrossPointNum)
15.         crossPoint ← validCrossoverPoint[crossPIndex]
16.         (child1, child2) ← crossOver(father, mother, crossPoint)
17.         population ← add(population, child1, child2)
18.     end if
19.     i++
20. end while
21. return population
```

Algorithm 3: two-point mutation operator

Input: *population, P*
Output: *population*

1. **while** $i < P$ **do**
2. $chromosome \leftarrow population[i]$
3. $chromLen \leftarrow \text{sizeof}(chromosome.character)$
4. $mutPoint1 \leftarrow \text{random}(1, chromLen)$
5. $mutPoint2 \leftarrow \text{random}(1, chromLen)$
6. **if** $chromosome.character[mutPoint1] \neq chromosome.character[mutPoint2]$ **then**
7. $newchromosome \leftarrow \text{swap}(chromosome, mutPoint1, mutPoint2)$
8. $population \leftarrow \text{add}(population, newchromosome)$
9. **end if**
10. i++
11. **end while**
12. **return** *population*

5. Model Validation and performance evaluation of GRPA

In this section, we first validate the analytical model of CFE described in section 3 with simulation in NS-3 and then evaluate the performance of the GRPA algorithm.

5.1 Model validation

We compare the experimental value and the theoretical value of the maximum delay. The experiment is conducted in NS-3. The theoretical value is calculated based on formula (21) and (22) in section 3. There are six cases corresponding to changing the rules placement solution for 10,000 synthetic flows.

5.1.1 Simulation setup

CFE is constituted by one hardware switch and two software switches. The length of every packet through CFE is 500 bytes. Thus, the capacities are 200 million packets per second (Mpps) and 10 Mpps respectively for 800Gbps hardware switch and 40Gbps software switch [8], which are the reasons that the *FlowTableLookupDelay* attributes of hardware switch and software switch are set 5 ms and 100ms respectively.

CFE forwards 10,000 synthetic flows concurrently. The sender of each flow uses a token bucket filter (TBF) to constrain the flow. The size of bucket buffer is set larger than the bucket size of the same TBF. Based on the traffic traces provided by the campus network of Beijing University of Posts and Telecommunications, the average packet rates of 10,000 concurrent flows at the same forward element are got, which are used as the average rates of TBFs. The burst size of each TBF is assumed to be the packet number that can be transmitted at the average rate over 500ms [28]. These flows are generated in terms of ON-OFF model. The ON time is a random number following an exponential distribution, while the OFF time is 500ms for the chance that token buckets are filled up.

Table 1. number of active software switches and average rates and burst sizes of the token buckets constraining the aggregated flows in six cases

		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Number of active software switches		2	2	3	3	4	4
F_0	ρ_0 (pps)	435596	381146	435596	381146	435596	381146
	σ_0 (packets)	217798	190573	217798	190573	217798	190573
F_1	ρ_1 (pps)	54453	81675	36302	54453	27225	40836
	σ_1 (packets)	27227	40838	18151	27227	13613	20418
F_2	ρ_2 (pps)	54446	81674	36298	54446	27230	40835
	σ_2 (packets)	27223	40837	18151	27223	13615	20418
F_3	ρ_3 (pps)			36299	54450	27198	40838
	σ_3 (packets)			18150	27225	13599	20419
F_4	ρ_4 (pps)					27246	40840
	σ_4 (packets)					13623	20420

Six simulation cases, which have individual flow allocations among channels of CFE, are constructed based on different rules placement solutions and the number of active slow channel (i.e. software switch). Two software switches are active in both case 1 and 2, three software switches are active in case 3 and 4, and four software switches are active in case 5 and 6. The sum of average rates corresponding the flows whose rules are placed into software switches is about 20% of the sum of average rates of all 10,000 flows in case 1, 3 and 5. This proportion is about 30% in case 2 , 4 and 6. **Table 1** presents the number of active software switches and the average rates and burst sizes of the token buckets constraining the aggregated flows in these six cases. As declared in section III, F_i denotes the aggregated flow passing through channel i in CFE, ρ_i and σ_i denote the average rate and burst size of the token bucket constraining F_i respectively.

5.1.2 Results

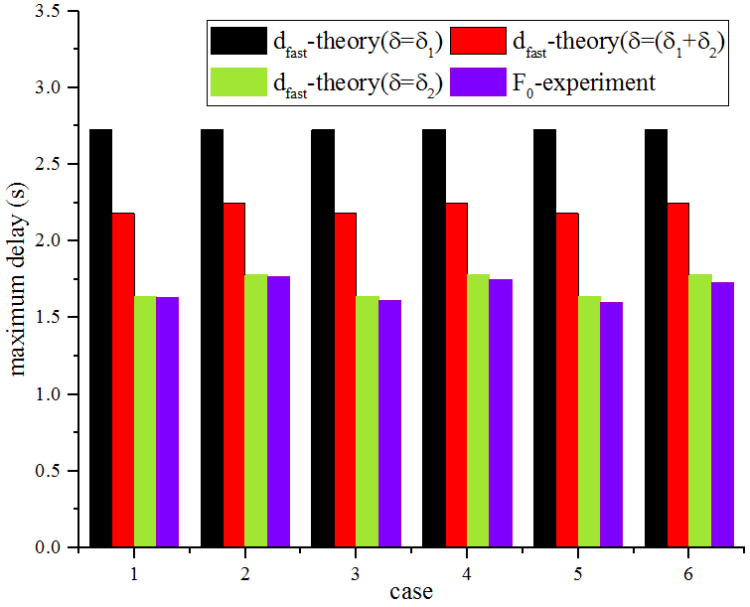
Fig. 3 depicts the experimental value of the maximum delay of any aggregated flow under six cases, and their three theoretical values. Three theoretical values of each aggregated flow under every case are obtained respectively when $\delta = \delta_1 = (\rho_0\sigma)/(m(2\rho - \rho_0))$, $\delta = \delta_2 = (\rho_0\sigma - \rho\sigma_0)/(m(2\rho - \rho_0))$, and $\delta = (\delta_1 + \delta_2)/2$. As explanted in section III, $\rho = \sum_{k=0}^{m} \rho_k$ and $\sigma = \sum_{k=0}^{m} \sigma_k$, where $m = 2$ in case 1 and 2 , $m = 3$ in case 3 and 4, and $m = 4$ in case 5 and 6.

Three results can be observed through the comparison among the results shown in **Fig. 3**. First, comparing the results of case 1, 3 and 5 with the results of case 2, 4 and 6 respectively, we found that not only the maximum delays of F_1 , F_2, F_3 and F_4 but also the maximum delay of F_0 increases with $\sum_{l=1}^{m} \rho_k$ and $\sum_{l=1}^{m} \sigma_k$. Thus, whether a rule is placed into the hardware switch of CFE depends not only on the CFE delay requirement of the flow

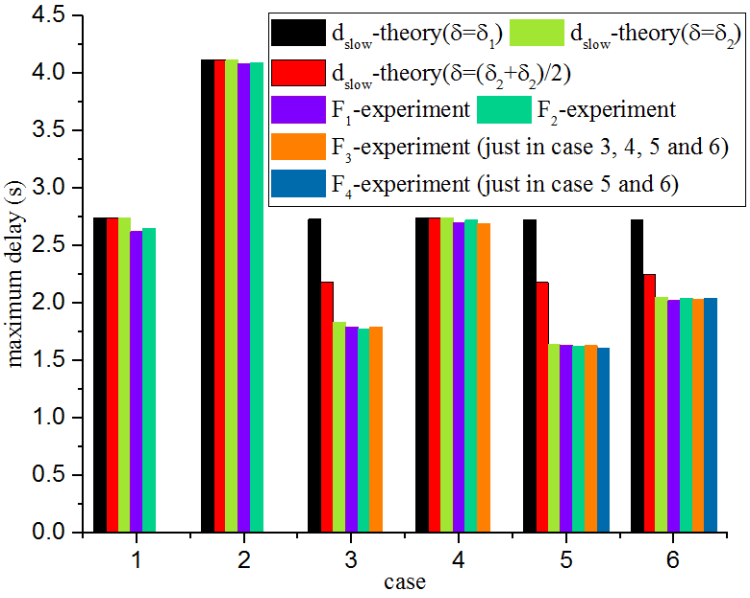
corresponding to this rule, but also on the effect exerted by this flow on the delays of other flows.

Second, the theoretical value of the maximum delay of any aggregated flow when $\delta = \delta_2$ is more precise approximation with the experimental value.

Lastly, the theoretical value of the maximum delay of F_1 , F_2, F_3 and F_4 is always tight approximation with the experimental value, and do not change with δ in case 1, 2 and 4. The reason is that the software switches are the bottlenecks of service providers to F_1, F_2, F_3 and F_4 in case 1, 2 and 4.



(a) F_0



(b) F_1 , F_2, F_3 and F_4

Fig. 3. experimental values and theoretical values of the maximum delays of the aggregated flows

5.2 Performance evaluation of GRPA

5.2.1 Benchmark algorithms

Delay based algorithms(DA): Because the delay bound caused by the fast channel is smaller than that caused by the slow channel in CFE, the priorities that the rules of the concurrent flows are placed in the hardware switch decrease with the CFE delay requirements of flows.

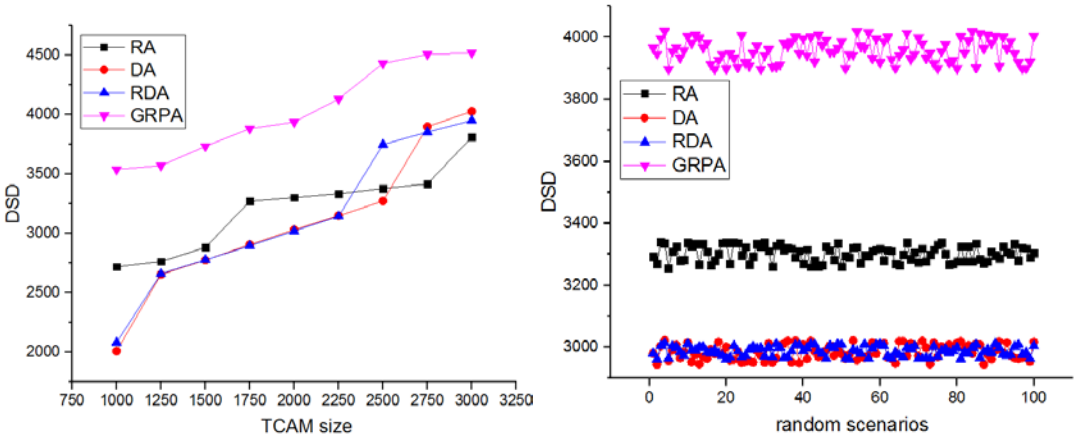
Rate based algorithm(RA): From the section 5.1.2, it is obtained that the bigger the sum of the average rates of the token buckets which constrain the flows pass through slow channel is, the higher delay bounds caused by the fast channel and the slow channel are. In RA, the priorities that the rules are placed in the hardware switch increase with the average rates of the token buckets which constrain the corresponding flows.

Rate-delay based algorithm (RDA): In this algorithm, the CFE delay requirement of the flow and the average rate of the token bucket which constrain the flow are both considered. The priority that the rule of the flow f is placed in the hardware switch increase with the value of ρ^f / l_f .

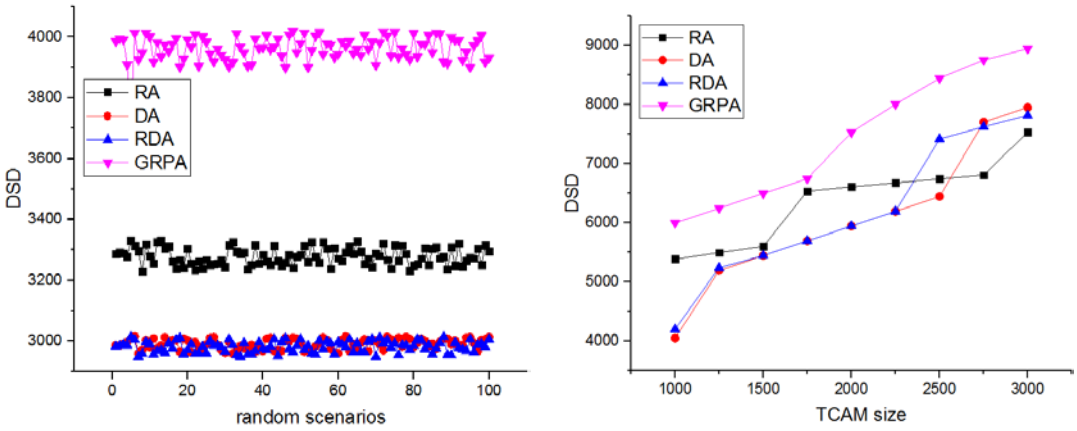
5.2.2 Simulations and results

In the simulations, the parameters of CFE and flows are presented as follows. There are four software switches in CFE. The capacity of hardware switch is 200Mpps, while that of software switch is 10Mpps. the average rates of 10,000 flows mentioned in section 5.1 are adopted again. The bust size of token bucket constraining each flow is still set as the number of packets that are generated by this flow in 500ms at the average rate. The CFE delay requirements of flows are chosen randomly from the interval [1ms, 10ms]. The weights of the flows are chosen randomly from the interval (0, 1).

In GRPA, the size of the initial population is 1000 and the crossover probability are 0.9. The rules placements solved by DA, RA and RDA are taken as three special chromosomes of the initial population. The end condition is that the iteration has been repeated 2000 or the best fitness remains same for 200 generations.



(a) DSD vs TCAM **(b)** DSD vs Flow Weight



(c) DSD VS the CFE delay requirements of flows **(d)** DSD vs TCAM ($\omega_f = 1$)

Fig. 4. comparison of four algorithms

Fig. 4(a) shows that the comparison of four algorithms under the case that the TCAM sizes are different while the flow weights and the CFE delay requirements of flows keep unchanged. **Fig. 4(b)** shows that the comparison of four algorithms under the case that the flow weights are 100 different uniform distributions while the TCAM sizes are always 2000 and the CFE delay requirements of flows keep unchanged. **Fig. 4(c)** shows that the comparison of four algorithms under the case that the CFE delay requirements of flows are 100 different uniform distributions while the TCAM sizes are always 2000 and the flow weights size keep unchanged. From **Fig. 4(a)**, **(b)** and **(c)**, it can be observed that GRPA can obtain higher DSD than the other three benchmark algorithms for any TCAM size, any uniform distribution of the CFE delay requirements of flows and any uniform distribution of the flow weights. **Fig. 4(d)** shows that the comparison of four algorithms under the case that is the same as the case corresponding to **Fig. 4(a)** except which the weights of all flows equal one. In this case, DSD indicates the number of flows of which the CFE delay requirements are satisfied. From **Fig. 4(d)**, it can be observed that GRPA guarantees the CFE delay requirements of more flows than the other three benchmark algorithms.

6. Conclusion

TCAM is a key enabler of hardware SDN switch to implement 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 that there is a lack of consideration on delay of rules placement in CFE, we propose an analytical model of 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. Based on the proposed analytical model, we present the GRPA algorithm to maximize the degree that the CFE delay requirements of flows are satisfied.

References

[1] OpenFlow 1.0.0 Specification [Online]. Article (CrossRef Link)

[2] M. Ku´zniar, P. Pereˇs´ıni, and D. Kosti´c, "What you need to know about sdn flow tables," in *Proc. of 16th Passive and Active Measurement*, pp.347–359, March 19-20, 2015. Article (CrossRef Link)

[3] A. Lazaris, D. Tahara, X. Huang, E. Li, A. Voellmy, Y. R. Yang, and M. Yu, "Tango: Simplifying sdn control with automatic switch property inference, abstraction, and optimization," in *Proc. of 10th ACM Conf. on Emerging Networking Experiments and Technologies*, pp. 199–212, December 02-05, 2014. Article (CrossRef Link)

[4] Tcams and openflow: What every sdn practitioner must know, [Online]. Article (CrossRef Link)

[5] M. Kobayashi, S. Seetharaman, G. Parulkar, G. Appenzeller, J. Little, J. Van Reijendam, P. Weissmann, and N. McKeown, "Maturing of openflow and software-defined networking through deployments," *Computer Networks*, vol. 61, pp. 151–175, March, 2014. Article (CrossRef Link)

[6] S. Banerjee and K. Kannan, "Tag-in-tag: Efficient flow table management in sdn switches," in *Proc. of IEEE Conf. on Network and Service Management (CNSM) and Workshop*, pp.109–117, November 17-21, 2014,. Article (CrossRef Link)

[7] 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. Article (CrossRef Link)

[8] N. Katta, O. Alipourfard, J. Rexford, and D. Walker, "Cacheflow: Dependency-aware rule-caching for software-defined networks," in *Proc. of ACM Symposium on SDN Research*, pp.6:1–6:12. March 14-15, 2016. Article (CrossRef Link)

[9] X. Cao, Y. Dong, and D. H. C. Du, "Synchronized multi-hop scheduling for real-time traffic on sdns," in *Proc. of 24th IEEE Conf. on Computer Communication and Networks*, pp.1–8, August 3-6, 2015,. Article (CrossRef Link)

[10] 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*, vol. 9, no. 4, p. 42, 2013. Article (CrossRef Link)

[11] X. N. Nguyen, D. Saucez, C. Barakat and T. Turletti, "OFFICER: A general optimization framework for OpenFlow rule allocation and endpoint policy enforcement," in *Proc. of IEEE Conference on Computer Communications*, pp. 478-486, April 26-May 1, 2015. Article (CrossRef Link)

[12] K. Kannan and S. Banerjee, "Flowmaster: Early eviction of dead flow on sdn switches," in *Proc. of International Conf. on Distributed Computing and Networking*, pp.484-498, January 4-7, 2014. Article (CrossRef Link)

[13] M. Klymash, B. Strykhalyuk, M. Beshley and T. Maksymyuk, "Research and development the methods of quality of service provision in Mobile Cloud systems," in *Proc. of IEEE International Black Sea Conference on Communications and Networking*, pp.160-164, June 5-8, 2014. Article (CrossRef Link)

[14] T. Maksymyuk, S. Dumych, O. Krasko and M. Jo, "Software defined optical switching for cloud computing transport systems," in *Proc. of 9th ACM International Conf. on Ubiquitous Information Management and Communication*, pp.1-2, January 08-10, 2015: Article (CrossRef Link)

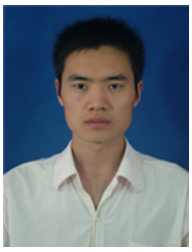
[15] M. Selmchenko, M. Beshley, O. Panchenko and M. Klymash, "Development of monitoring system for end-to-end packet delay measurement in software-defined networks," in *Proc. of 13th IEEE International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science*, pp. 667-670, February 23-26, 2016. Article (CrossRef Link)

[16] K. Phemius and M. Bouet, "Monitoring latency with openflow," in *Proc. of 9th IEEE Conf. on Network and Service Management*, pp.122-125, October 122-125, 2013. Article (CrossRef Link)

[17] N. Mehravari, *Queueing Theory*. John Wiley and Sons, Inc., 2001. Article (CrossRef Link)

[18] J. L. Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*, Springer-Verlag, 2001. Article (CrossRef Link)

[19] Y. Jiang, "Network calculus and queueing theory: Two sides of one coin: Invited paper," in *Proc. of 4th International ICST Conf. on Performance Evaluation Methodologies and Tools*, pp. 37:1–37:12, October 20-22, 2009. Article (CrossRef Link)
[20] F. Ciucu and J. Schmitt, "Perspectives on network calculus: no free lunch, but still good value," in *Proc. of ACM conf. on Applications, technologies, architectures, and protocols for computer communication*, pp. 311–322, August 13-17, 2012. Article (CrossRef Link)
[21] J. Y. L. Boudec, "Application of network calculus to guaranteed service networks," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 1087–1096, 1998. Article (CrossRef Link)
[22] A. Lazaris, D. Tahara, X. Huang, E. Li, A. Voellmy, Y. R. Yang, and M. Yu, "Tango: Simplifying sdn control with automatic switch property inference, abstraction, and optimization," in *Proc. of 10th ACM Conf. on Emerging Networking Experiments and Technologies*, pp. 199–212, December 02-05, 2014. Article (CrossRef Link)
[23] E. J. Rosensweig and J. Kurose, "A network calculus for cache networks," in *Proc. of IEEE INFOCOM*, pp. 85–89, April 14-19, 2013. Article (CrossRef Link)
[24] J. L. Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*, Springer-Verlag, 2001. Article (CrossRef Link)
[25] C.S. Chang, *Performance guarantees in communication networks*, Springer Science & Business Media, 2012. Article (CrossRef Link)
[26] J. Y. Leung, *Handbook of scheduling: algorithms, models, and performance analysis*, CRC Press, 2004. Article (CrossRef Link)
[27] L D. Whitley, "The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best," in *Proc. of ICGA*, pp.116-123, 1989. Article (CrossRef Link)
[28] ACL and QoS Command Reference [Online]. Available: Article (CrossRef Link)



Qinglei Qi received the B.S. and M.E. degrees from Henan University of Technology, Zhengzhou, China in July 2008 and 2011, respectively. He is currently a Ph.D candidate at State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include IP QoS and Software-Defined Networking.

Wendong Wang received the B.S. and M.E. degrees from Beijing University of Posts and Telecommunications (BUPT), Beijing, China in 1985 and 1991, respectively. He is a full professor at BUPT. His research interests include the next-generation network architecture, IP QoS and mobile Internet.

Xiangyang Gong received the B.S. and M.E. degrees from Xi'an Jiaotong University(XJTU), Xi'an, China in 1992 and 1995, respectively, and the Ph.D. degree in communication and information system in 2011 from Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is a full professor at BUPT. His research interests are IP QoS, video communications, the next-generation network architecture, and mobile Internet.

Xirong Que received the B.E. and M.E. degrees from Beijing University of Posts and Telecommunications (BUPT), Beijing, China in 1993 and 1998, respectively. She is currently an associate professor at BUPT. Her research interests include the next-generation network architecture, IP QoS and mobile Internet.