RateSheriff: Multipath Flow-aware and Resource Efficient Rate Limiter Placement for Data Center Networks

Songshi Dou[⋆], **Yongchao He**[†], **Sen Liu**[‡], **Wenfei Wu**[§], **& Zehua Guo**[⋆]

[⋆] Beijing Institute of Technology
[†] Tsinghua University
[‡] Fudan University
[§] Peking University

IEEE/ACM IWQoS'23 (June 20, 2023)

Outline

Outline

Research Background

Emerging cloud services and applications

- **Different QoS requirements**
  - Latency-sensitive
    - Web services and video streaming
  - Throughput-intensive
    - Hadoop
  - Both latency and throughput
    - AR/VR, virtual gaming, and tactile Internet
- **Limited network resource**
  - Competing for the bottleneck bandwidth
  - Leading to performance fluctuation
- **Possible solution - rate limiter**
  - Limiting the flow rate
  - Realizing performance isolation

Research Background

Programmable switch-based rate limiter

Compared with server-based rate limiter

- **Easy consistency control**
  - Without consuming extra server resources and getting access to end-hosts
- **High precision and throughput**
  - Without enduring the request of scheduling or queuing resources

Emerging in-network computing

- **Many in-network applications**
  - Key-value caching (*e.g.*, NetCache)
  - Coordination service (*e.g.*, NetChain)
  - Gradient aggregation (*e.g.*, ATP)
- **The limited memory resource in programmable switches**

Outline

Motivation and Challenges

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Limitations of existing solutions

- **Multipath flows cannot be precisely limited**
  - Servers can be virtualized into many VMs controlled by different tenants
  - Single-path and multipath flows could co-exist in DCNs
  - Existing designs mainly consider single-path flows
- **The control plane solution for rationally placing rate limiters in DCNs is missing**
  - The impact of varying flow rate on network performance
  - The limited switch memory resource

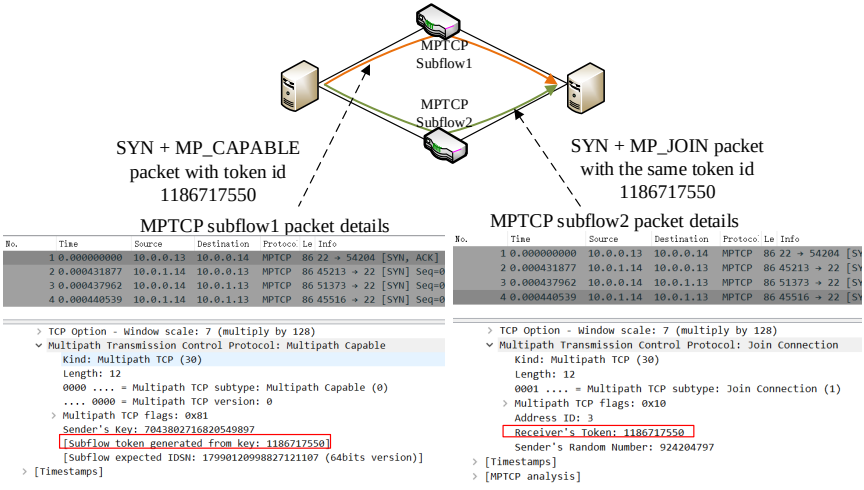
Challenges

- **How to identify MPTCP flows**
  - The solution should accurately distinguish two types of flows
  - The identification process should be lightweight
- **How to realize performance and resource efficiency in the entire network**
  - The different placement of rate limiters in DCNs affects the performance of rate limiting

Outline

Outline

MPTCP Flow Identification



Multipath flow identification.

Outline

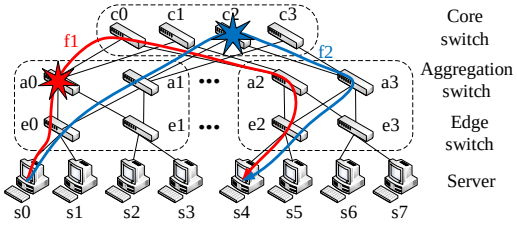
Placement Considerations

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Examples

The MPTCP flow m is expected to be limited to 200 Mb/s

- Case 1: Assume f_1's rate reaches 150 Mb/s, and f_2's rate reaches 70 Mb/s
  - If f_1 and f_2's limiters are set to 100 Mb/s, m can only work at 170 Mb/s
  - If f_1 and f_2's limiters are set to 150 Mb/s, m's rate becomes 220 Mb/s
- Case 2: The two subflows do not incur bandwidth utilization waste in the network[1]



Placing rate limiters for MPTCP flows.

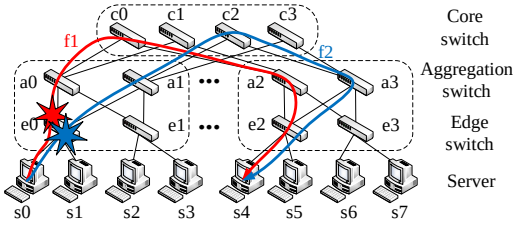
[1] E. Song *et al.*, "A cloud-scale per-flow backpressure system via FPGA-based heavy hitter detection", in ACM SIGCOMM'21 Posters.

Placement Considerations

Examples

The MPTCP flow m is expected to be limited to 200 Mb/s

- Case 1: Assume f_1's rate reaches 150 Mb/s, and f_2's rate reaches 70 Mb/s
  - If f_1 and f_2's limiters are set to 100 Mb/s, m can only work at 170 Mb/s
  - If f_1 and f_2's limiters are set to 150 Mb/s, m's rate becomes 220 Mb/s
- Case 2: The two subflows do not incur bandwidth utilization waste in the network[1]



Placing rate limiters for MPTCP flows.

[1] E. Song *et al.*, "A cloud-scale per-flow backpressure system via FPGA-based heavy hitter detection", in ACM SIGCOMM'21 Posters.

Outline

Outline

Rate limiter placement location constraint

- The rate limiter can only be placed at the switch which the flow's path traverses

$$x_{ij} \leq \alpha_{ij}, \forall i \in [1, N], \forall j \in [1, K]. \tag{1}$$

Rate limiter placement number constraint

- Each flow's rate limiter can be only deployed at one switch

$$\sum_{i=1}^{N} x_{ij} = 1, \forall j \in [1, K]. \tag{2}$$

MPTCP subflow constraint

- For each MPTCP subflow from the same MPTCP connection, the rate limiter can be only placed at the same switch

$$p_{jj'} \leq \sum_{i=1}^{N} x_{ij} * x_{ij'}, \forall j, j' \in [L+1, K], j' \neq j. \tag{3}$$

Memory usage constraint

- We use u_i to denote the memory usage of switch s_i

$$u_i = \sum_{j=1}^{L} x_{ij} * m_j, \forall i \in [1, N]. \tag{4}$$

- The used memory at each switch should not exceed the switch's memory capacity M_i

$$u_i \leq M_i, \forall i \in [1, N]. \tag{5}$$

Outline

Objective functions

- To maximize the overall performance benefit of rate limiter deployment

$$obj_1 = \sum_{j=1}^{K} \sum_{i=1}^{N} x_{ij} * r_j * \beta_{ij}.$$

- To let switches have balanced memory utilization

$$obj_2 = h = \max(u_i), \forall i \in [1, N]. \quad (6)$$

Outline

Resource Efficient Rate Limiter Placement problem

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Problem formulation

$$\max_{x,h} \sum_{j=1}^{L} \sum_{i=1}^{N} x_{ij} * r_j * \beta_{ij} - \lambda * h \tag{P}$$

subject to

Eqs. (1)(2)(3)(4)(5)(6)

$$h \geq 0, x_{ij} \in \{0,1\}, \forall i \in [1,N], \forall j, j' \in [1,K], j' \neq j.$$

Problem reformulation

- The high complexity comes from Eq. (3) since two binary variables are multiplied
- We propose to pre-place rate limiters for each MPTCP flow at the edge close to senders

Outline

Heuristic Solution

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

Heuristic algorithm - RateSheriff

Step 1: obtaining the linear programming relaxation solution

- Generating $\bar{X} = \{x_k, k \in [1, N * L]\}$ by solving the linear programming relaxation of the RERLP problem
- Sorting the results in the descending order

Step 2: finding feasible placement

- Testing potential deployment based on the descending order of their probabilities
- Placing rate limiters for the rest of flows by relaxing the constraint of balancing memory usage

Algorithm 1 Heuristic solution

Input: F, N, L, P_j, m_j, M_i, U_i, β_{ij}, Cap;
Output: $\mathcal{X}$;

```
1: X = ∅;
2: Sort all flows in the set F in the descending order of their
   flow rate difference before and after the rate limiter;
3: Generate Avg_Mem;
4: // set the memory utilization capacity of each switch to
   ensure the memory usage is balanced.
5: if max(U_i) ≥ Avg_Mem then
6:     Cap = max(U_i);
7: else
8:     Cap = Avg_Mem;
9: end if
10: // test potential placement based on the descending order
    of their flow rate difference.
11: for f_j0 ∈ F do
12:     for s_i0 ∈ P_j0 do
13:         // rate limiter is placed at switch s_i0 for flow f_j0.
14:         if U_i0 + m_j0 ≤ Cap and M_i0 ≥ m_j0 then
15:             M_i0 = M_i0 − m_j0, U_i0 = U_i0 + m_j0;
16:             F ← F \ f_j0, X ← X ∪ (i0, j0);
17:             break;
18:         end if
19:     end for
20: end for
21: // place rate limiters for the rest of flows by relaxing the
    constraint of balancing memory usage.
22: if F! = ∅ then
23:     for f_j0 ∈ F do
24:         for s_i0 ∈ P_j0 do
25:             if M_i0 ≥ m_j0 then
26:                 M_i0 = M_i0 − m_j0, U_i0 = U_i0 + m_j0;
27:                 F ← F \ f_j0, X ← X ∪ (i0, j0);
28:                 break;
29:             end if
30:         end for
31:     end for
32: end if
33: return X;
```

Outline

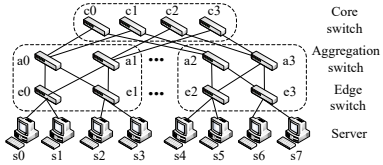
Evaluation

Simulation setup

- 3-layer 8-pod fat-tree network
- 100K TCP flows randomly between end-hosts
- The benefit of placing a rate limiter is set proportionally to the distance for the flow's source server to the switch that is placed with the limiter

Comparison algorithms

- Nearest
- MultiPath-Oblivious (MP-Oblivious)
- Optimal
- Memory-Balancing (Mem-Balancing)
- Without rate limiter
- RateSheriff



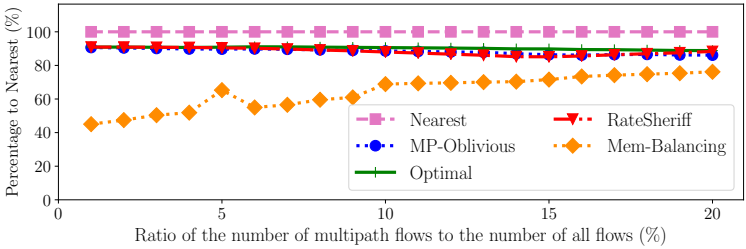
Fat-tree topology.

Evaluation

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

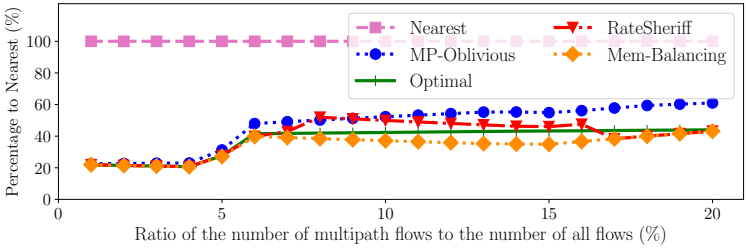
Overall benefit performance

- The higher, the better
- RateSheriff: 46% higher than Mem-Balancing at most



Memory balancing performance

- The lower, the better
- RateSheriff: improving memory balancing performance by up to 79%



- More results are in the paper

Outline

Summary

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

New ovservations

- We identify two limitations of existing programmable switch-based rate limiter designs
  - They cannot identify and limit multipath flows
  - They are lack of the consideration to rationally place rate limiters for control plane

New problem and solution

- We formulate new problem for placing rate limiters, which is MINLP
- To reduce the complexity, we **reformulate the problem** and **provide rigorous proof** of its complexity

Good performance

- We **propose a heuristic solution named RateSheriff** to efficiently solve the proposed problem
- We evaluate the performance of RateSheriff under a typical DCN

Thank you for your attention!

Q&A

songshidou@hotmail.com