Proportional Bandwidth Sharing Using Bayesian Inference in SDN-based Data Centers

Zheng Luo

December 6, 2016

Quick look

This article:

- *targets* delivering rates proportional to weights of subscribed flows on TCP/IP stack. ($r_i \propto w_i$)
- *by* probabilistically dropping L2 packets
- *where* dropping rate of $flow_i$ is determined by:
  - the weight (given as input by a tenant) (w_i)
  - current system load l
  - its contribution to the system load [posterior probability] ($p(flow_i|l)$)
    - How to compute $p(flow_i|l)$?

$$Drop_f \propto \frac{p(flow_i|l)}{w_i} \tag{1}$$

Predicting $p(flow_i|l)$

By Bayes' theorem:

$$p(f|l) \propto p(l|f)p(f) \tag{2}$$

$p(f)$ By sampling packets: $p(f) = packets_f/packets_{all}$

- Smoothing: $p(f) = \alpha p(f) + (1-\alpha)e(f)$

$p(l|f)$: Assume normal distribution$N(\mu, \sigma)$. At each sampling, l is known, so μ, σ could be updated

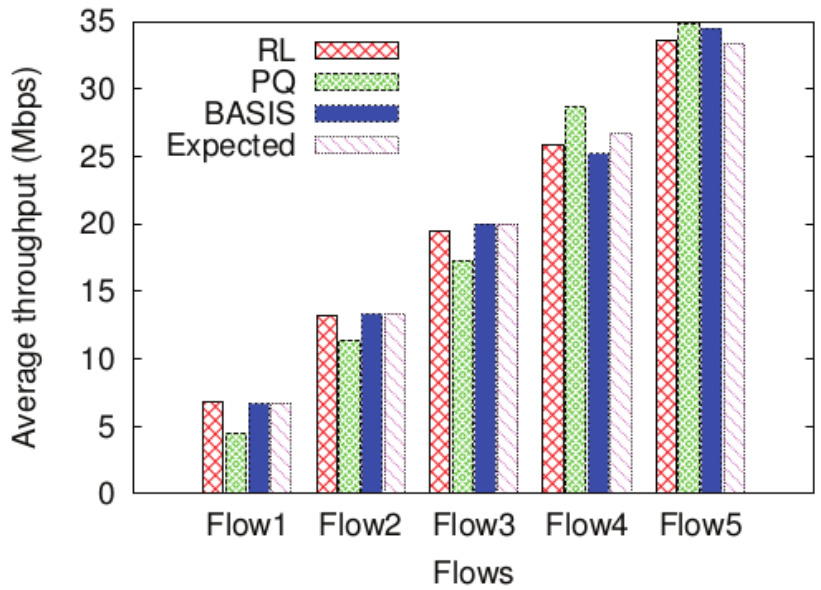
Algorithm

```
buffer = []
When packet incomes:
    if current_load > sampling_threshold: # Sampling
        if rand_with_prob(sampling_rate):
            buffer.append(packet)
        if buffer.full():
            for packet_in_buf in buffer:
                update(p_f, packet_in_buf.flow)
                update(p_l_given_f, packet_in_buf.fl
                    current_load)
            buffer.clear()
    if current_load > drop_threshold: # Dropping
        drop_rate = compute(p_f, p_l_given_f, packet
            current_load)
        if rand_with_prob(drop_rate):
            drop(packet)
```

Performance Study

Link capacity = 100Mbps, TCP

Flow ID	Source Rate (Mbps)	Flow Weight	Expected throughput (Mbps)
Flow1	15	1	6.67
Flow2	20	2	13.33
Flow3	25	3	20
Flow4	30	4	26.67
Flow5	35	5	33.33



Thanks for listening

Proportional Bandwidth Sharing Using Bayesian Inference in SDN-based Data Centers, Purnima Murali Mohan, Dinil Mon Divakaran, Mohan Gurusam, IEEE ICC 2016 - Next-Generation Networking and Internet Symposium