



Contribution ID: 170

Type: **not specified**

Traffic policing in eBPF: applying token bucket algorithm

Thursday, November 15, 2018 12:20 PM (20 minutes)

eBPF-based traffic policer as a replacement* of Hierarchical Token Bucket queuing discipline.

The key idea is two rate three color marker (rfc2698) algorithm, which inputs are committed and peak rates with the corresponding burst sizes and the output is a color or category assigned to a packet. There are conforming, exceeding, violating categories. An action is applied to violating category - either drop or dscp remark. Another action may optionally be applied to exceeding category.

Close-up of eBPF implementation**. Write intensiveness is a cornerstone: an update of available tokens is required on each packet, as well as tracking of time. Naive implementation and its exposure to data races on multi-core processors system. A problem of updating both timestamp and the number of available tokens atomically. Slicing the timeline into chunks of the size of burst duration as a solution for races, mapping each packet into its chunk, so there is no need in updating global timestamp. Two approaches of storing timeline chunks: bpf LRU hash map and a block of timeline chunks in bpf array. Circulating over a block of timeline chunks. Proc and cons of the latter approach: lock-free with bpf array as the only data structure used vs. increased amount of locked memory.

Combining several policers. Linear chain of policers instead of hierarchy. Passing a packet over the chain. Dealing with bandwidth underutilization when first K policers in a chain conform a packet and K+1 rejects. Commutative property of chained policers. Interaction with UDP and TCP. TCP reacting on drop by changing congestion window which affects the actual rate.

- Note, that it's a replacement, not the alternative: eBPF based implementation doesn't assume putting packets into queues. ** Since the action is per packet, eBPF program should be attached to tc chain, it doesn't work with cgroups.

Presenter: KARTSEVA, Julia (Facebook)

Session Classification: BPF MC