The paper introduced a new approach to controlling the network interference in datacenter networks, called qJump, which concentrate their effort on datacenter network rather than normal internet.

Since the throughput-intensive and latency-sensitive applications may share the same route in the network, means the switch, router or other network equipment. Throughput-intensive applications may consume most of the network resource, which caused the latency-sensitive ones suffering from the delay and influenced its performance(page 2).

This approach introduced a concept called network epoch, representing the worst delay that one packet may suffer before being transferred. And in the qJump approach, author wish to ensure the latency bound being guaranteed, so it become twice the value of normal network epoch, which is used to limit the flow rate in the network. (page 3) Author also combined priority values with network epoch, small network epoch means high network priority, on the contrary, bigger network epoch represent lower priority. And qJump also restrict the relationship between priority and network epoch: User will have a low latency, but low throughput network. Or a high throughput, and maybe highly delayed one(page 4).

So, with a application implementing qJump, when its packets arrive to the export of current environment, packets with higher priority will be send first than those ones have lower priority. However, higher priority means lower buffer size. If the application sends more data than the buffer on its priority allows, the packets will be dropped, this is how qJump enforce the rule between throughput and network delay(page 5).

As my concern, this new approach is similar to other software-based application, easy to be understand, and also easy to be implemented. According to the evaluation result displayed, qJump also have pretty good effect on balancing network flow efficiency and latency control(page 6).

On the other hand, I don’t think just drop the packet directly when application sent more data than it is allowed. Especially in TCP protocol, if a packet is lost, then TCP protocol enter the fast recover mode, and packet traffic in the network will be cut into half. If the qJump also drop packet except packet lost in normal traffic, it will definitely cause more trouble for the network utilization. One solution mentioned in the paper that put the application who send to much data traffic into a lower level is a pretty good idea. What’s more, in order to make the balance, qJump forced low latency tolerance applications to have less network traffic resources, and packets of network flow intense applications will have lower priority than the first one. But what happen when if some application need both high network flow and low network delay? For instance, let’s divide requirement of speed and delay into three different level. Applications need level 3 speed would have level 1 delay priority, and level1 speed are for those application need level3 latency tolerance. But if an application require level3 speed and level2 delay priority, than the latency requirement of this application wouldn’t be satisfied.

Considering about the points above, I believe for further investigation, maybe they can consider of not just implement this approach to datacenter network, but also develop a new version for normal Internet environment. But they need to avoid some disadvantages like mentioned above.