

Hash-Cast: A Dark Corner of Stochastic Fairness

Background

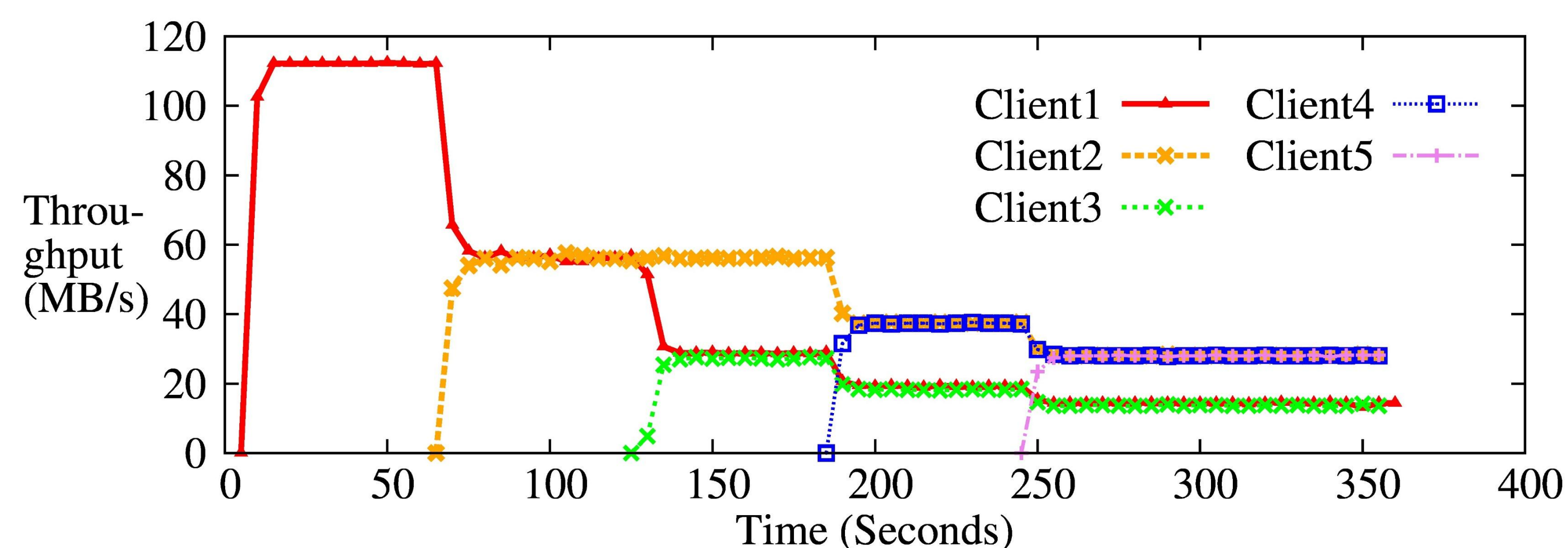
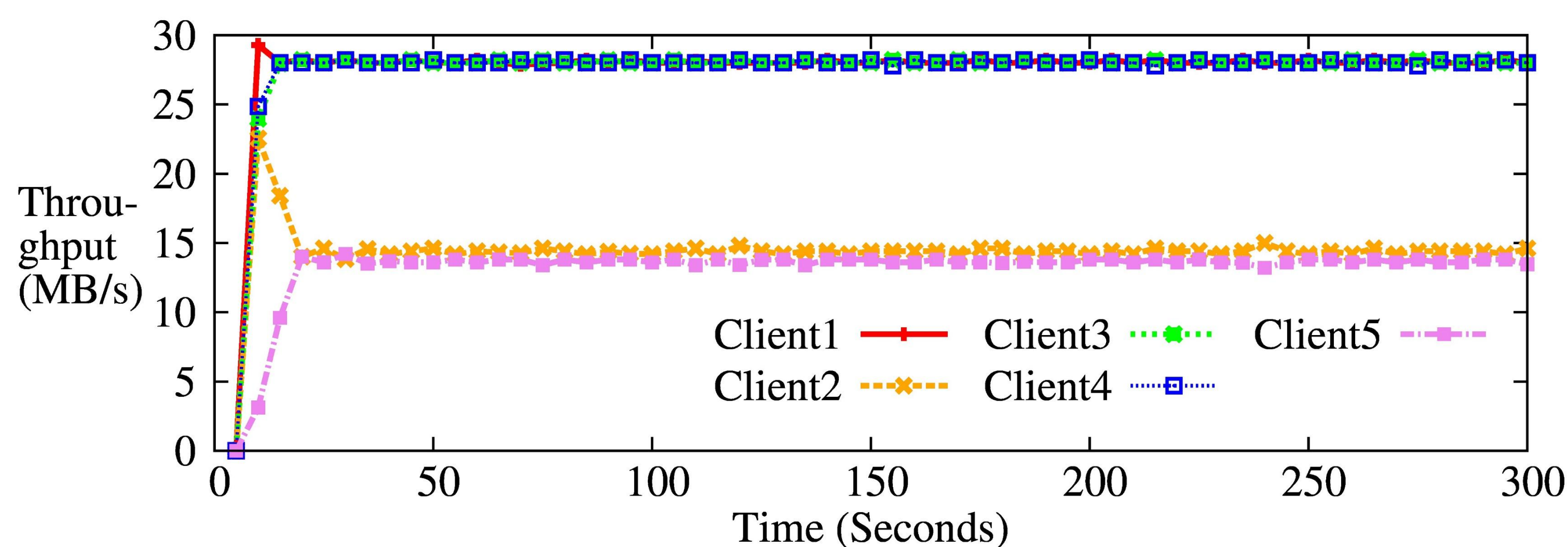
1

- ❖ We benchmarked NFSv4.1 on Linux:
 - Clients sequentially read a 20GB file via NFS
 - 64KB I/O size
- ❖ Experimental setup:
 - 6 identical machines: 1 server and 5 clients
 - Six-core Intel Xeon 2.66G CPU, 64 GB RAM
 - Broadcom 1GbE NIC
 - CentOS 6.4, 2.6.32-358el6 and vanilla 3.12.0
 - RAID-0 with two 15K RPM drives

Winner-Loser (WL) Pattern

2

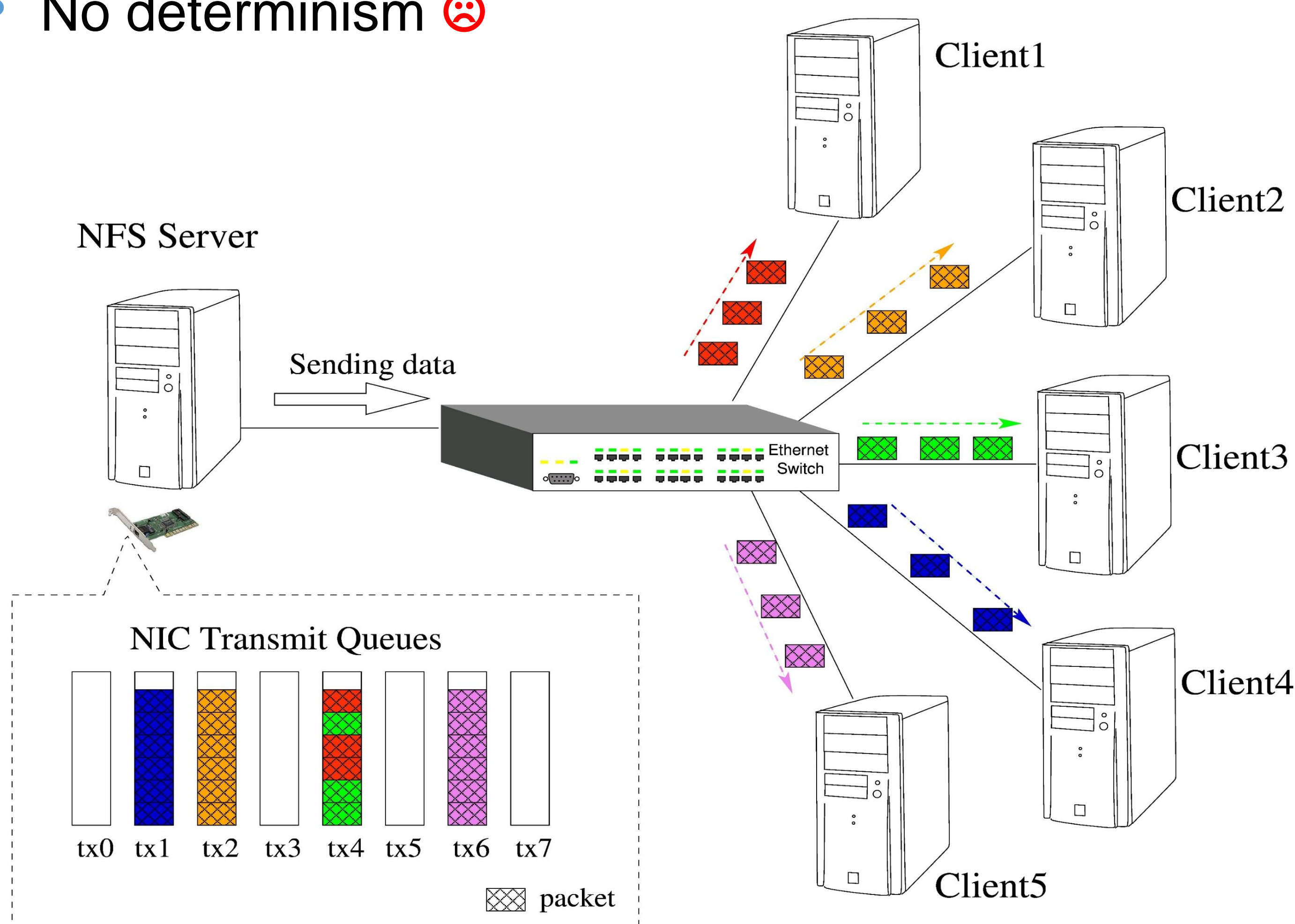
- ❖ WL pattern has three characteristics:
 1. Two clusters of clients: winners & losers
 2. Winner-loser throughput ratio: 2 (often) or 3 (seldom)
 3. WL consistency: yes within one run, no across runs



Hash-Cast

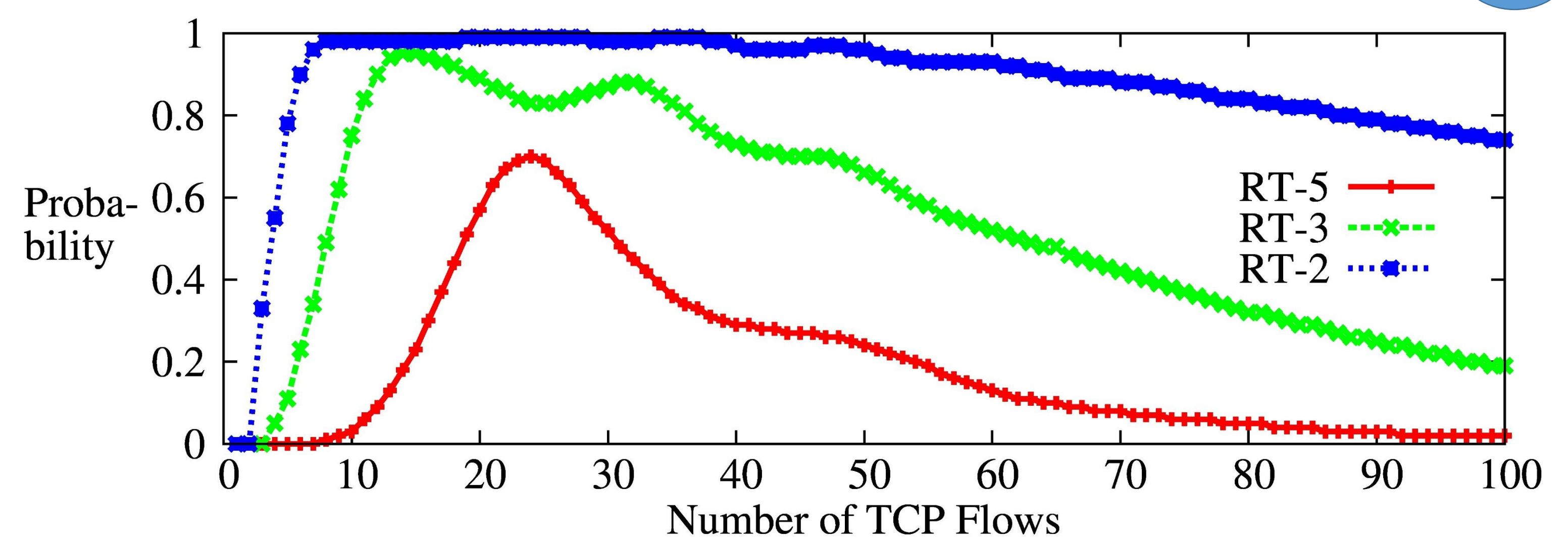
3

- ❖ Multi-queue NICs: offer scalability, improve virtualization.
- ❖ Hashing: active TCP flows → NIC queues
 - Simplicity and efficiency 😊
 - Stochastic fairness (uniformity of values) 😊
 - No uniformity with a small number of buckets ☹️
 - No absolute fairness guarantee ☹️
 - No determinism ☹️



Probability of Hash-Cast Occurring

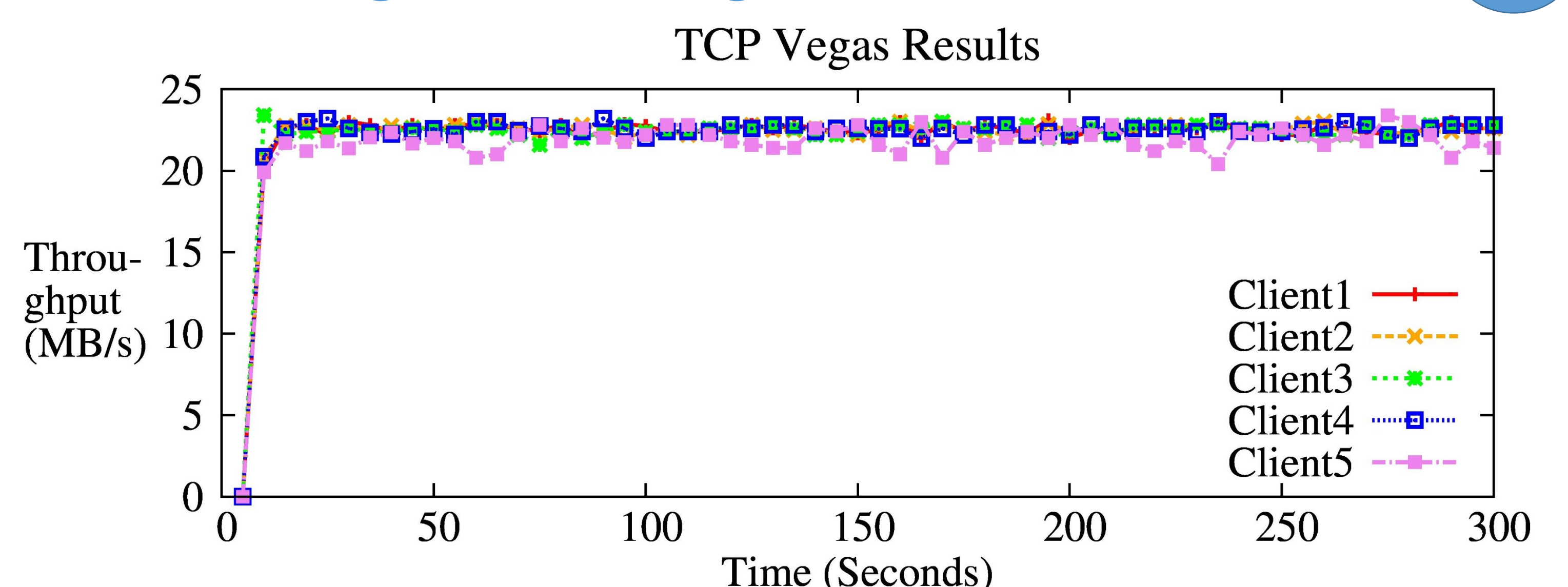
4



- ❖ Ratio Threshold (RT): highest thput./lowest thput.
- ❖ Hash-Cast occurs with high probability in typical scenarios (with 10 to 50 active TCP flows).

TCP Congestion Algorithm Matters

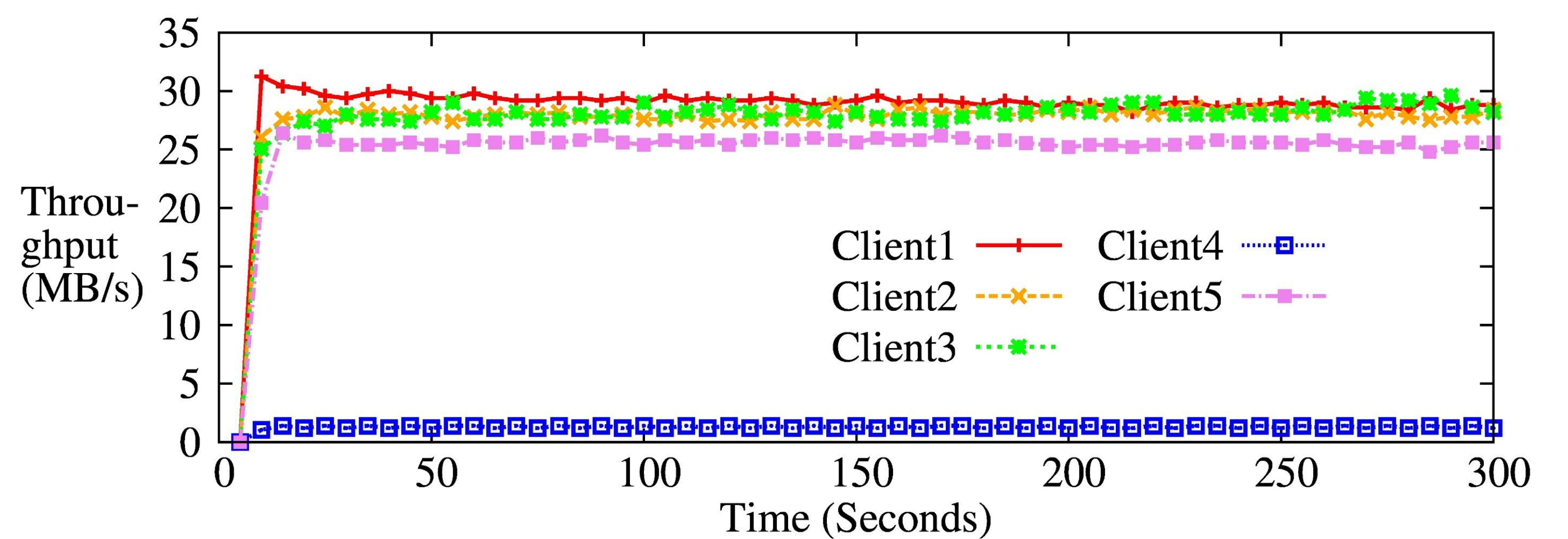
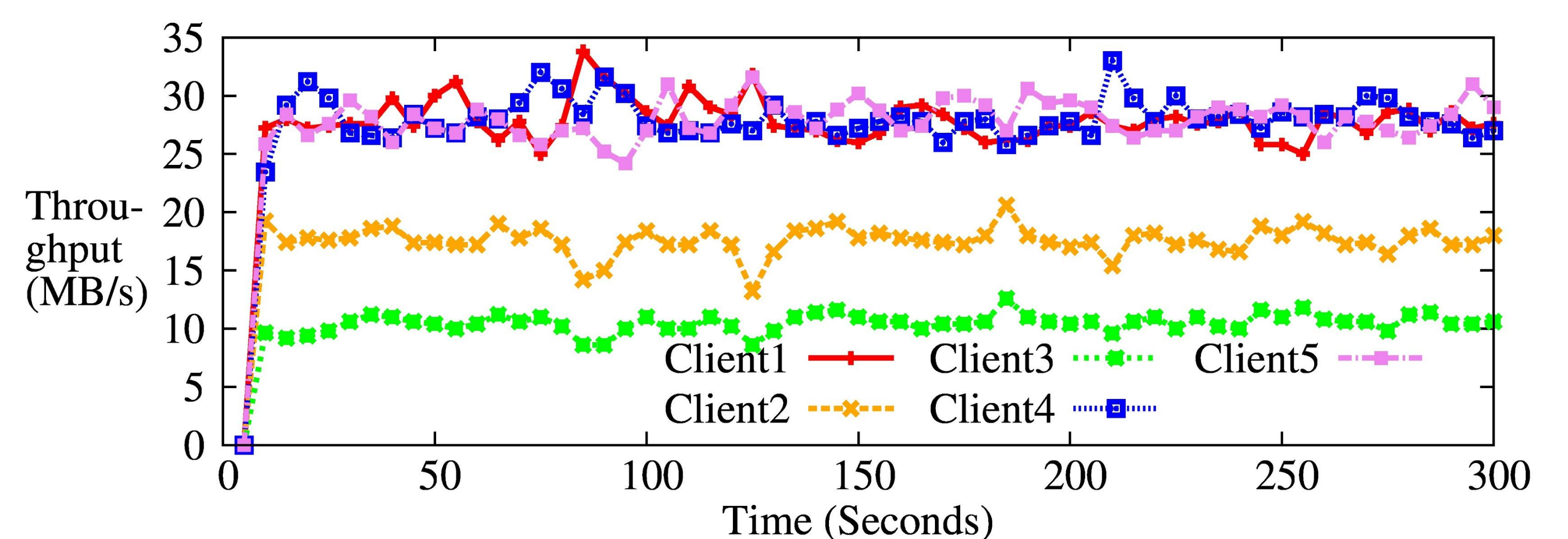
5



- ❖ TCP Vegas does not show WL pattern.
- ❖ TCP Vegas vs. Cubic (*cwnd* in packets: 26 vs. 1900.)
- ❖ Hash-Cast is exacerbated by *Bufferbloat*.

Hash-Cast Still Exists in Linux 3.12

6



- ❖ Hash-Cast still exists and becomes more complex.

Conclusion

7

- ❖ Hash-Cast
 - Occurs in the presence of multi-queue NIC
 - Causes unfairness (WL) with high probability
 - Happens in TCP: affects services hosting data-intensive outgoing TCP flows, e.g., NFS, file server
- ❖ Future work:
 - Determine scheduling among physical queues
 - Design and implement explicit load balancing
 - Fit multi-queue NIC into Linux traffic control