



TAALK: Server-Load Aware Network Load Balancing



Alaleh Azhir^{1,2,3}, Tony Yang¹, Lucy Zhang¹, Kristin Yim¹, Angelo Olcese¹

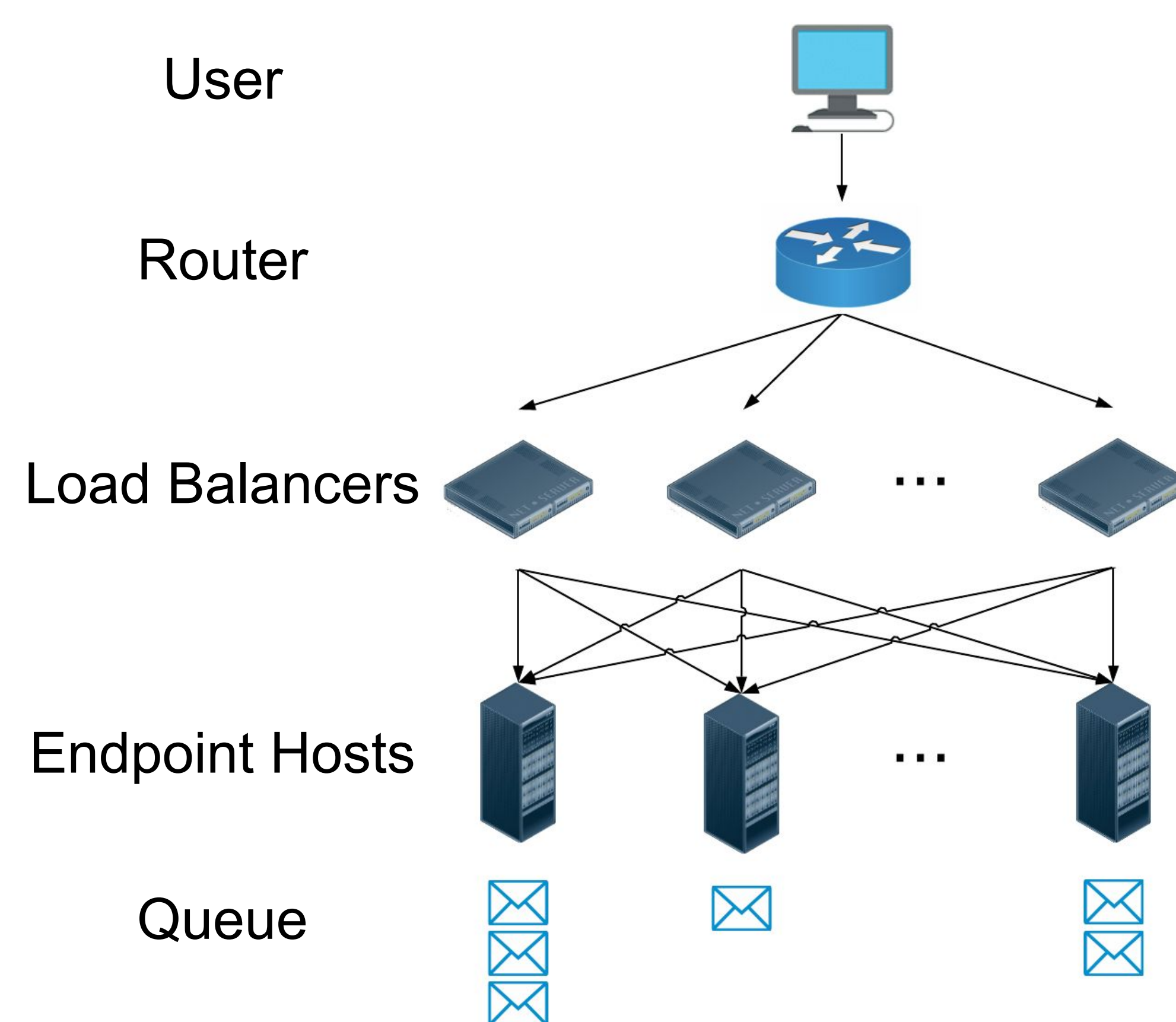
1. Department of Computer Science, Johns Hopkins University, Baltimore, Maryland, USA
2. Department of Biomedical Engineering, Johns Hopkins University, Baltimore, Maryland, USA
3. Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, Maryland, USA

1. Background

TAALK is a system designed to improve server load balancing by considering the endpoint servers' load when making initial routing decisions. This improves previous efforts in this space, such as consistent hashing methods, that do not consider server loads leading to high tail latency.

2. Questions

➤ What is a Load Balancer?

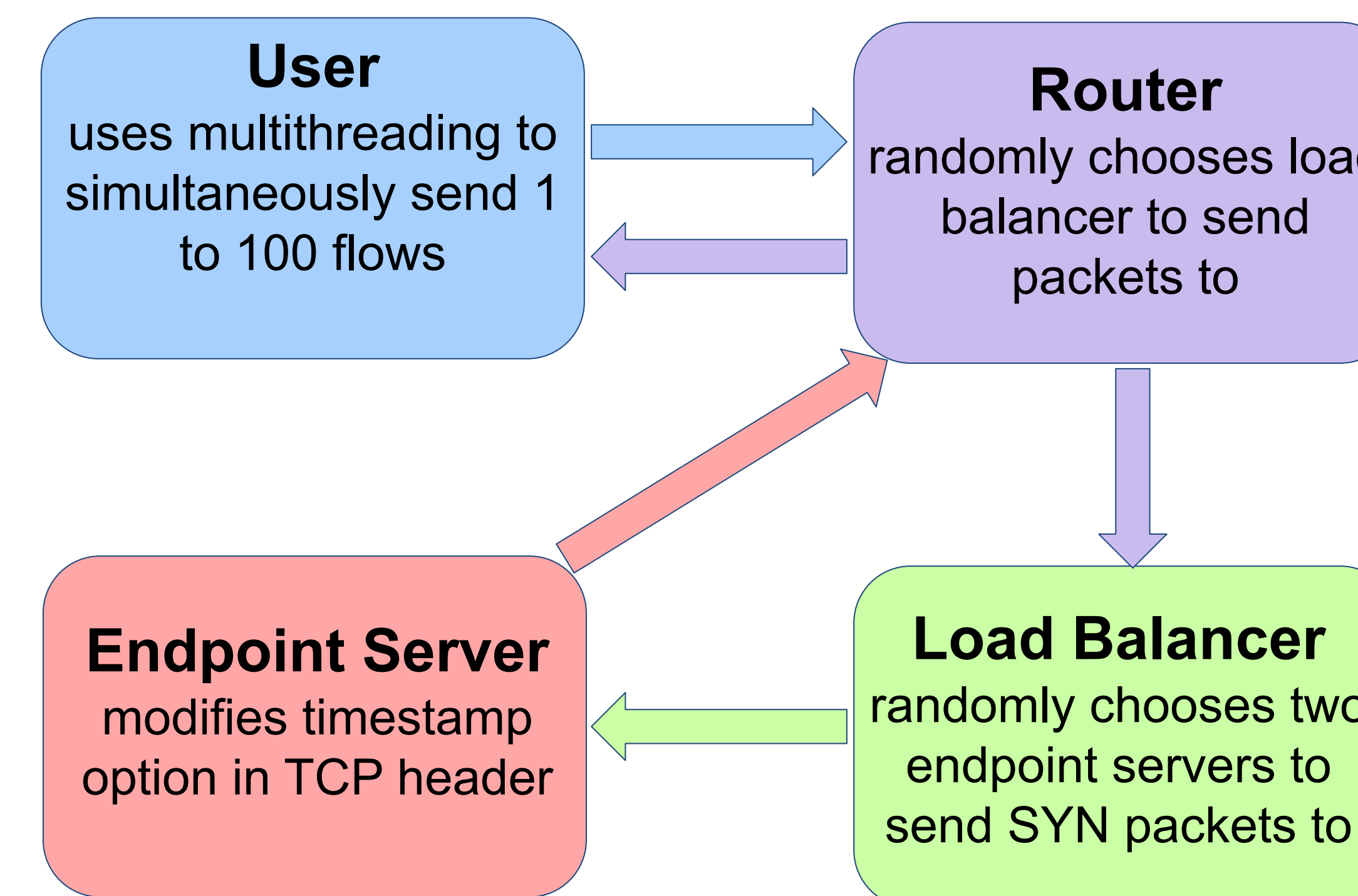
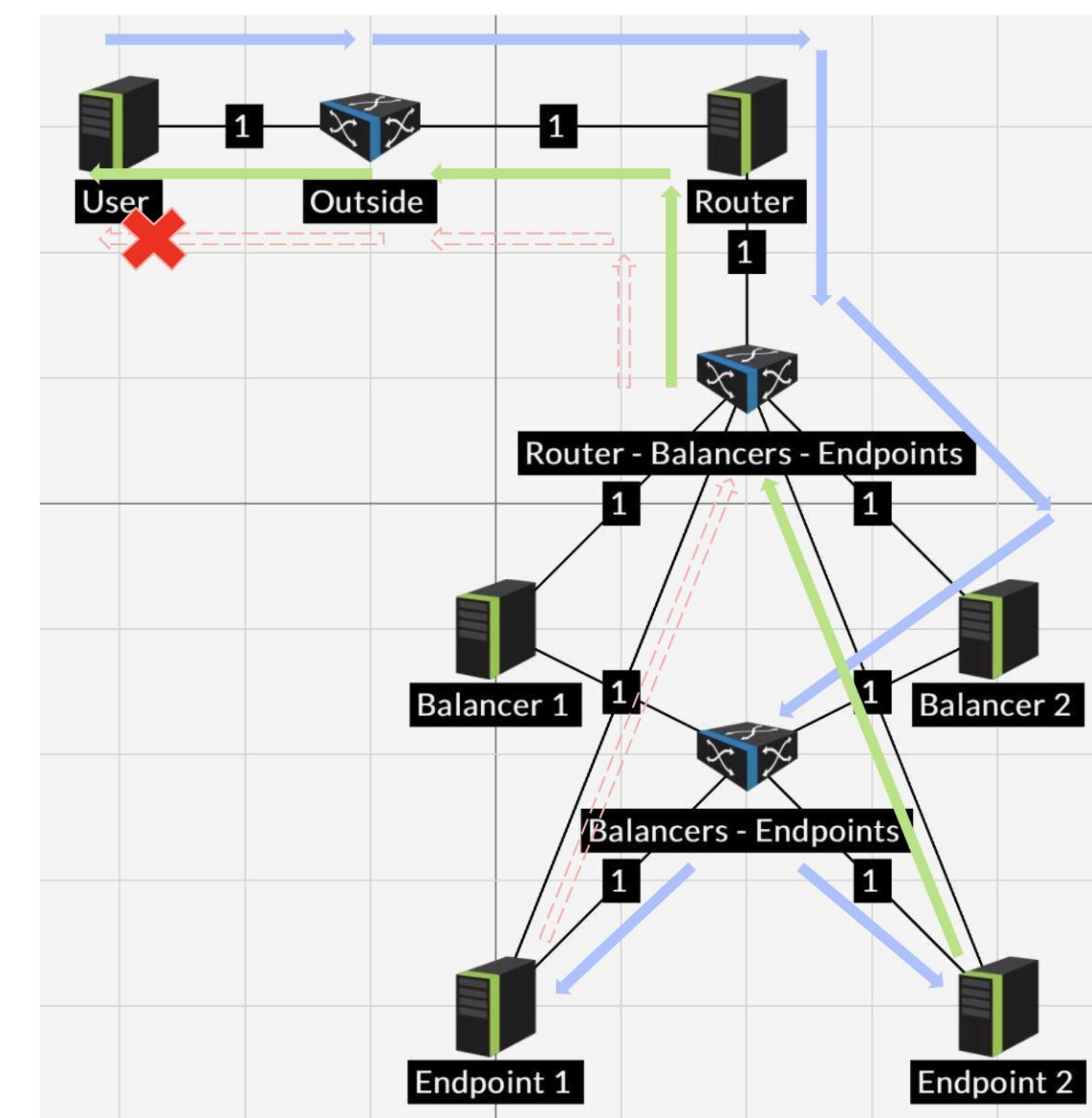


➤ What properties do we want?

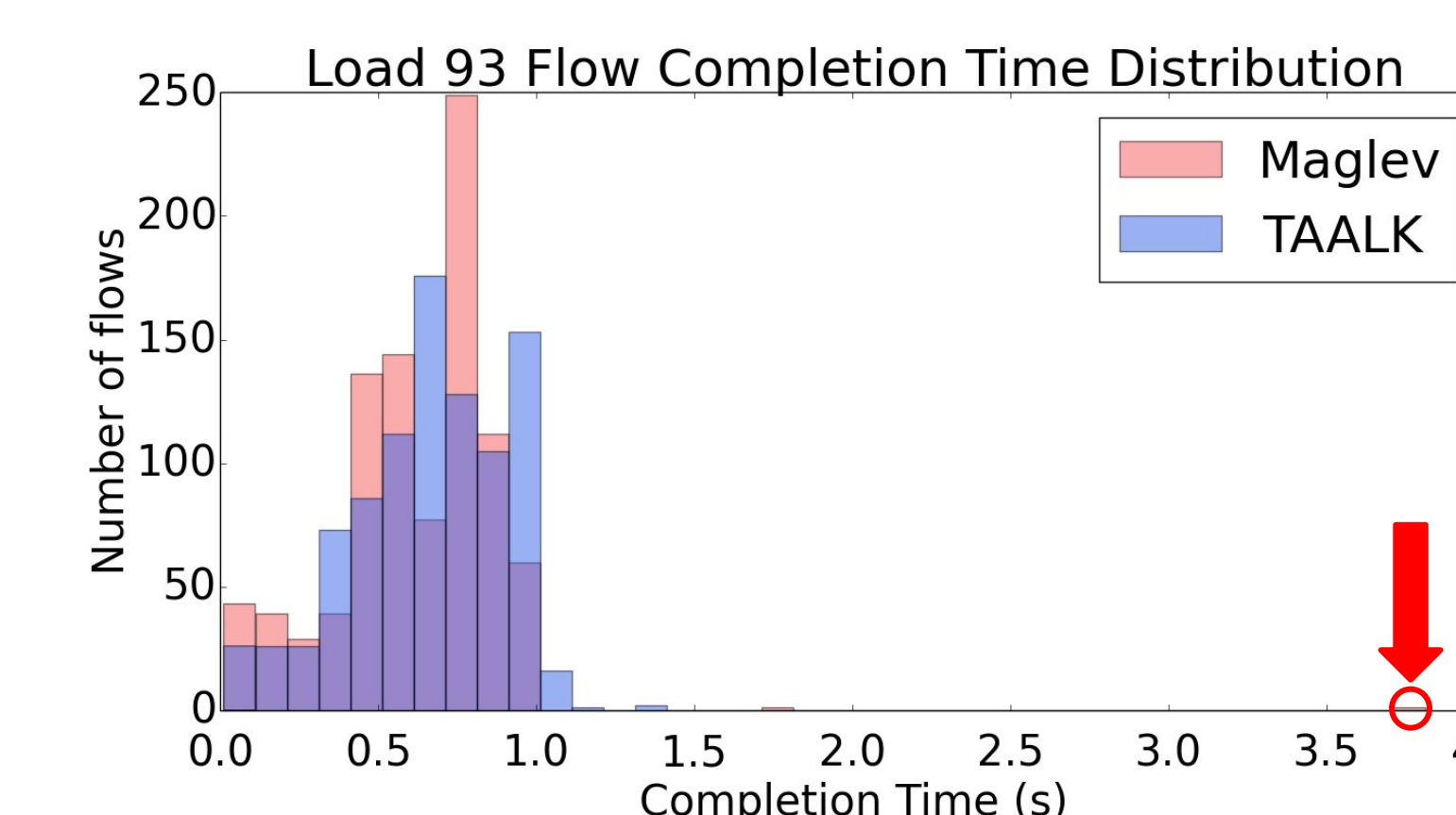
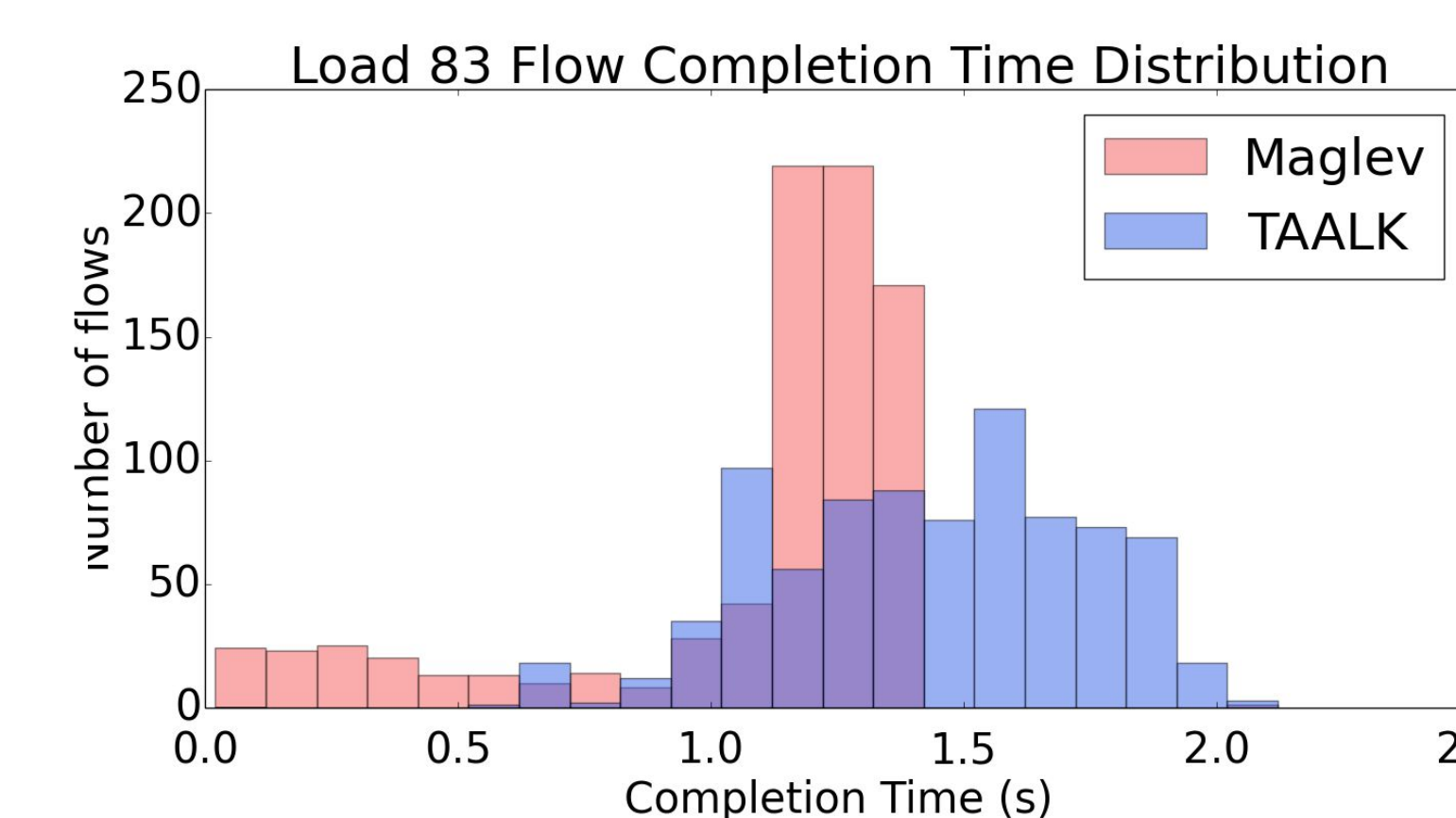
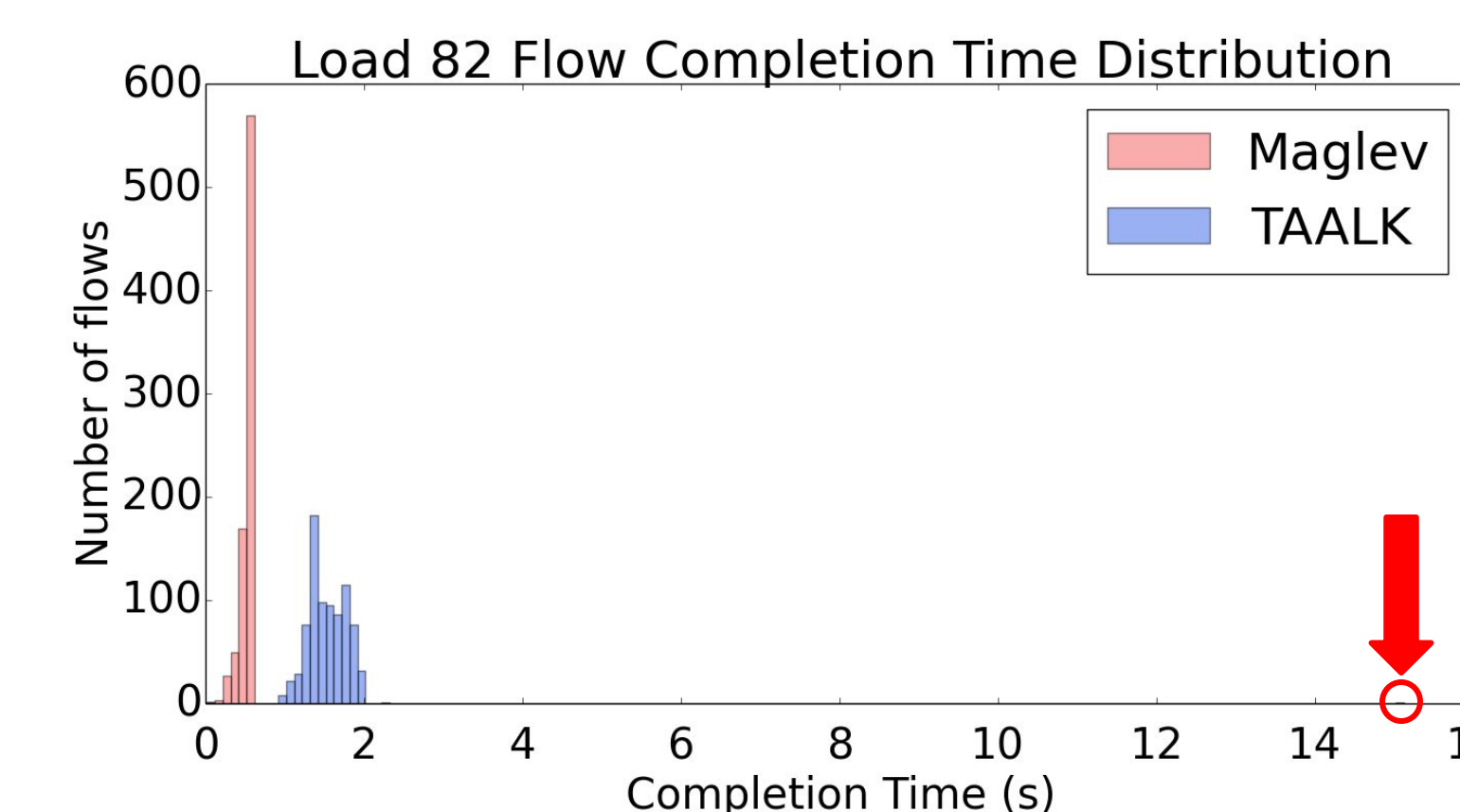
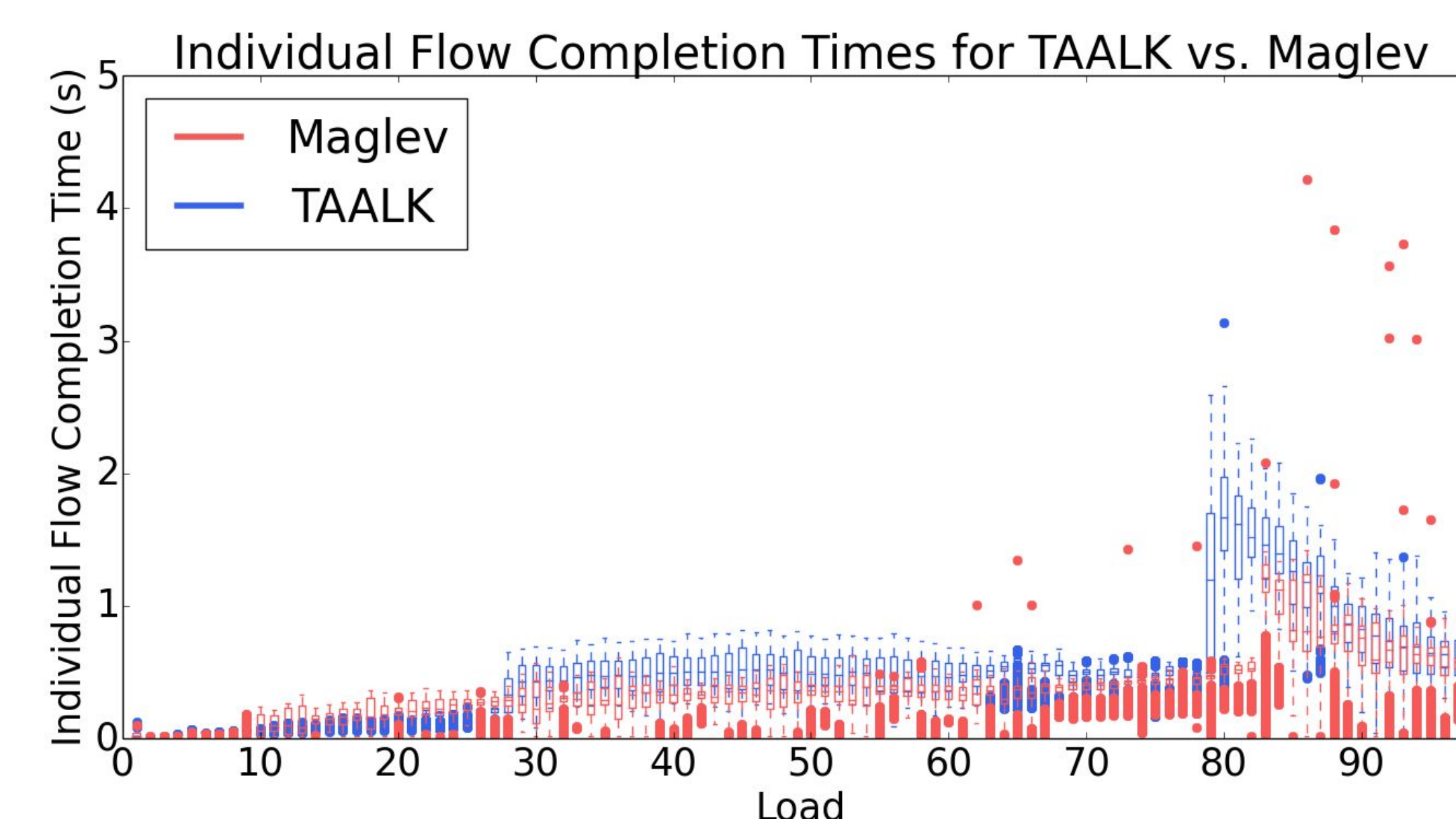
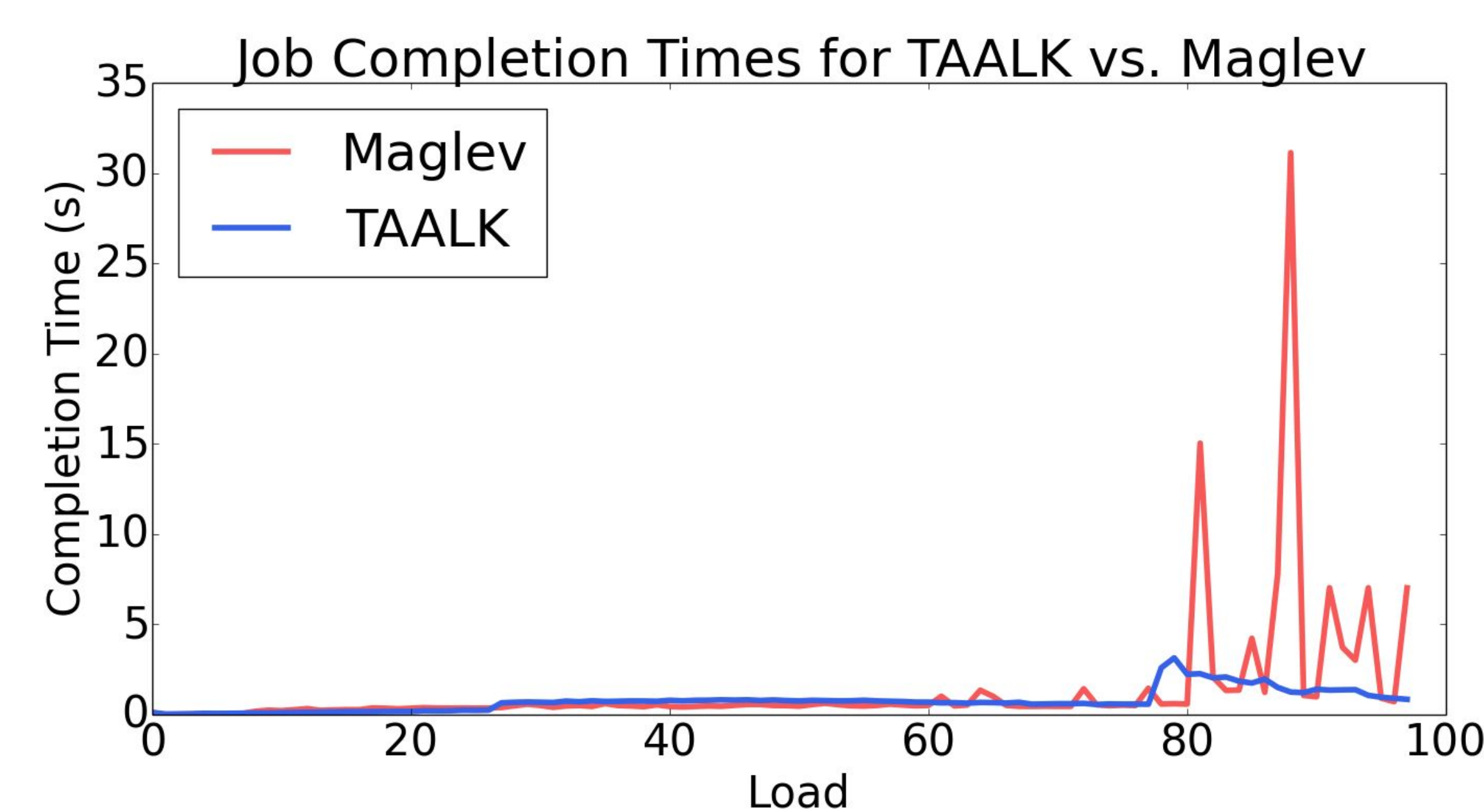
- Consistency - packets of the same flow go to the same endpoint server
- Low tail latency - no outliers with high completion time
- Low network overhead
- Low load on balancers - outgoing packets bypass load balancer

3. Architecture

➤ CypherPath



4. Results



5. Conclusion

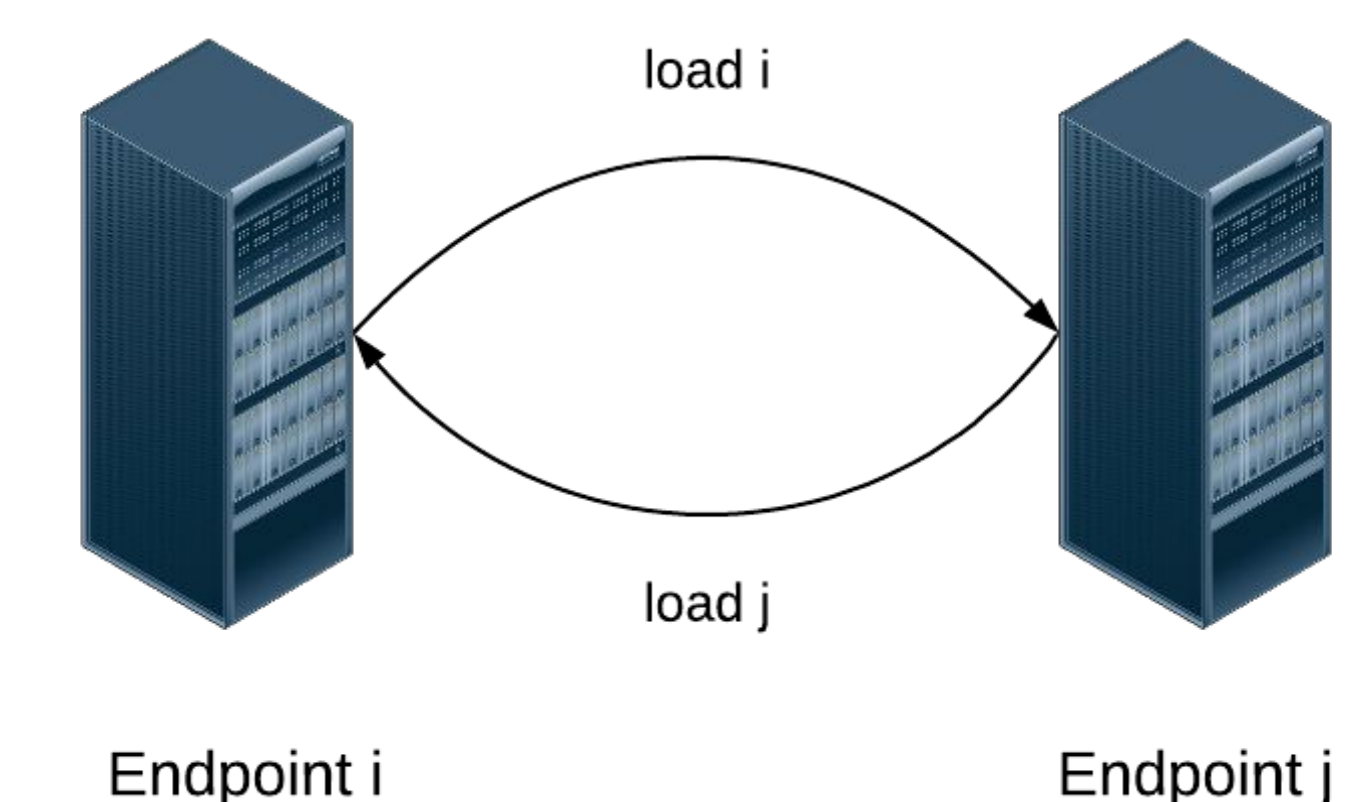
TAALK	Maglev
Server-load Aware	No awareness of servers' load
Chooses two possible endpoints and sends SYN packet to both.	Chooses one server randomly
Duplicated SYN packets and multiple Syn+Acks high network overhead	No network load added

➤ Why TAALK

- Low tail latency

➤ Future Work

- Endpoint servers communicate load with each other
- Scaling to larger topologies
- Different traffic simulations



6. References

- [1] A10 Networks AX Series. <http://www.a10networks.com>.
- [2] Cypherpath. (2019). <https://www.cypherpath.com/>.
- [3] Dean, Jeffrey, and Luiz André Barroso. "The tail at scale." Communications of the ACM 56.2 (2013): 74-80.
- [4] Eisenbud, Daniel E., et al. "Maglev: A fast and reliable software network load balancer." 13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16). 2016.
- [5] INET Framework. (2019). <https://inet.omnetpp.org/>.
- [6] Kemp. <http://www.kemptechnologies.com>.
- [7] Mininet. (2019). <http://mininet.org/>.
- [8] OMNeT++ Discrete Event Simulator. (2019). <https://omnetpp.org/>.
- [9] Patel, Parveen, et al. "Ananta: Cloud scale load balancing." *ACM SIGCOMM Computer Communication Review*. Vol. 43. No. 4. ACM, 2013.
- [10] V. Jacobson, R. Braden, D. Borman, TCP Extensions for High Performance, RFC Editor, 1992
- [11] West, M. and S. McCann, "TCP/IP Field Behavior", RFC 4413, DOI 10.17487/RFC4413, March 2006