

PostMan: Rapidly Mitigating Bursty Traffic by Offloading Packet Processing

Panpan Jin¹, Jian Guo¹, Yikai Xiao¹, Rong Shi², Yipei Niu¹,
Fangming Liu^{*1}, Chen Qian³, Yang Wang²

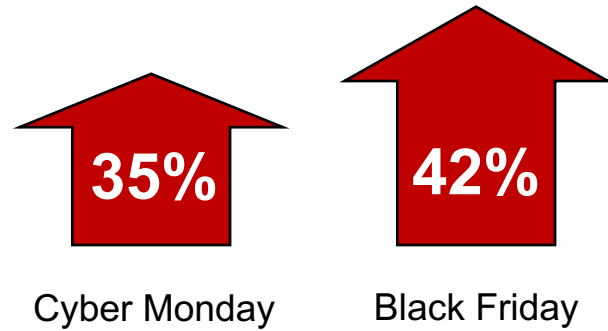
¹OpenCloudNeXt Group, Huazhong University of Science and Technology

²The Ohio State University

³University of California Santa Cruz

OpenCloudNeXt Group: <http://grid.hust.edu.cn/fmliu>

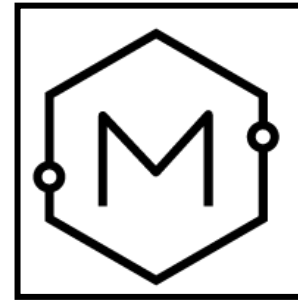
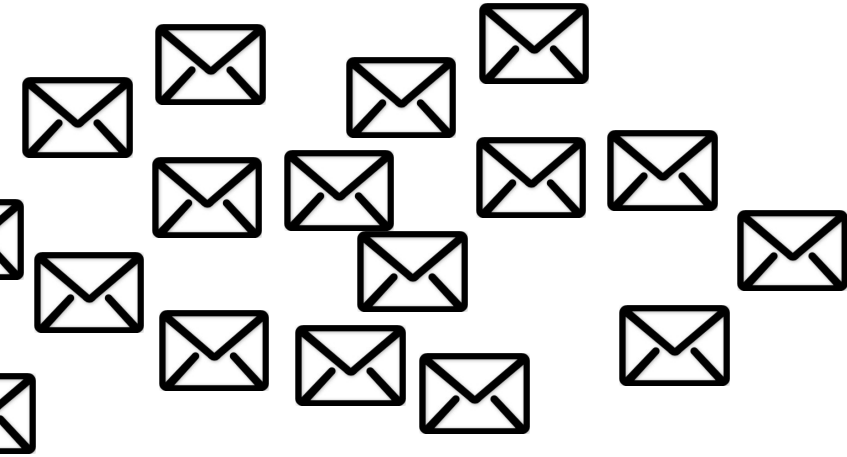
Bursty traffic is a headache!



Increase of
conversation
rate in **24h**

- ✓ **1,480,000,000** deals in total
- ✓ **42,000,000** queries/s at peak
- ✓ **256,000** deals/s at peak

Bursty traffic is a headache!



Memcached Servers

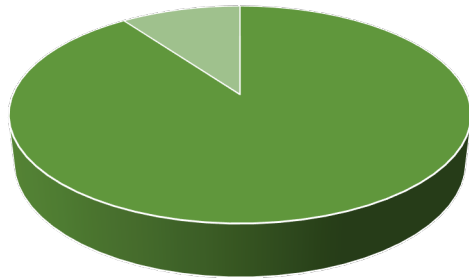


Large volume



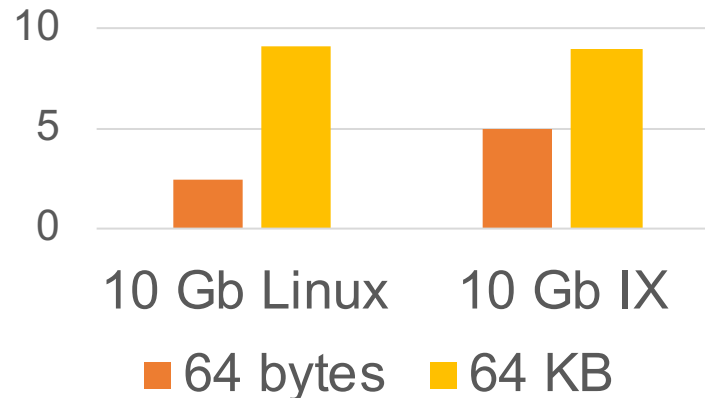
Short duration

Payload size breakdown



- [0, 31 bytes]
- (31 bytes, 41 bytes]

Packet processing throughput (Gbps)

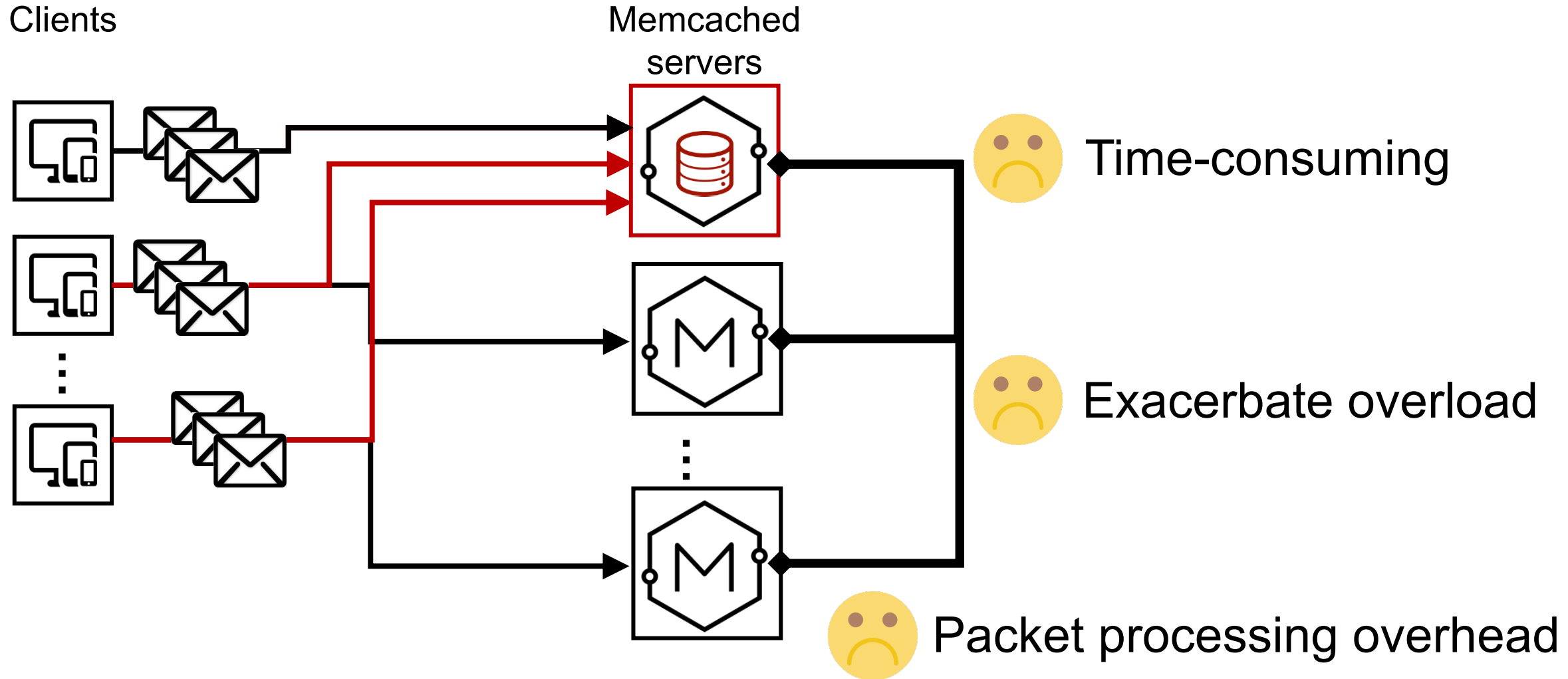


Small packets

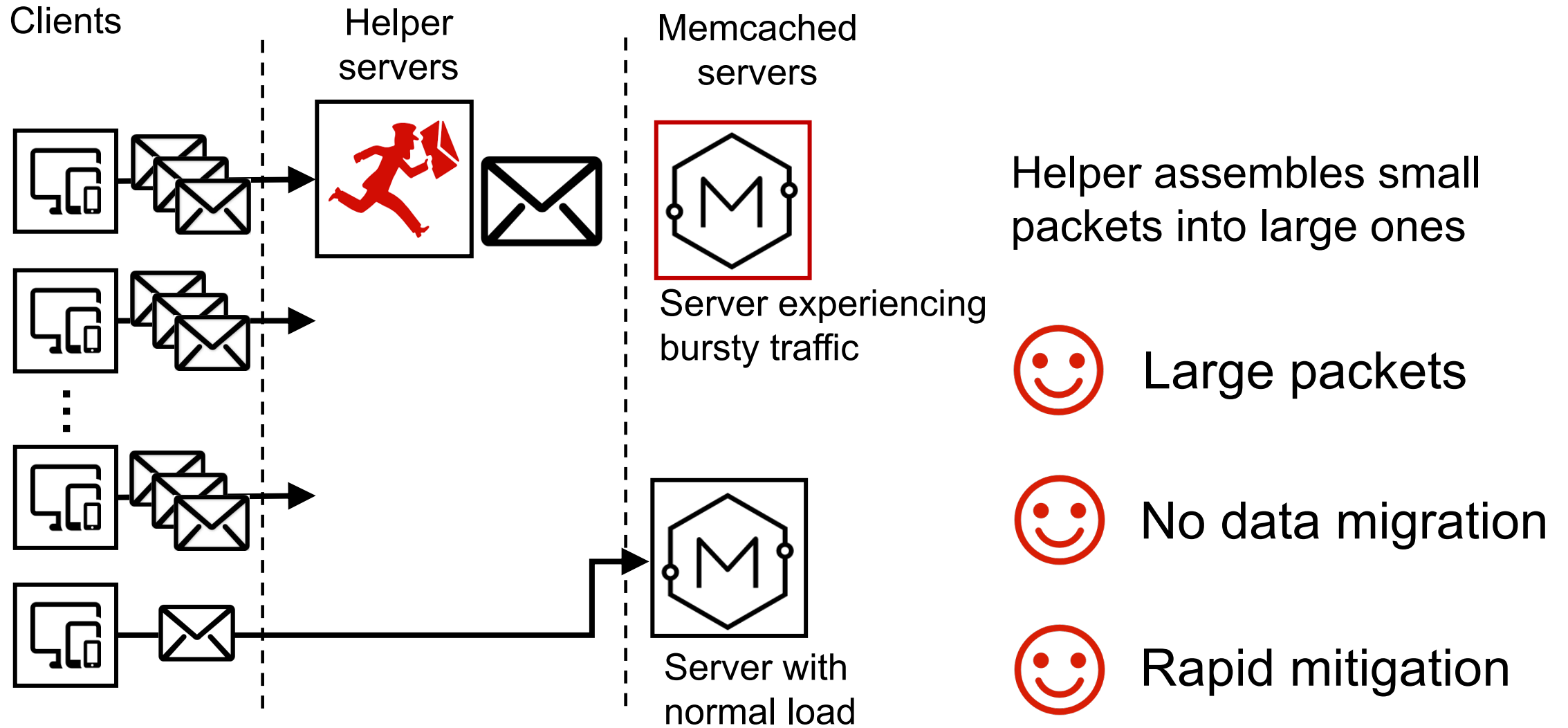


Severe overhead

Traditional remedy: migrating hot data for load balancing



PostMan: batching and offloading on demand

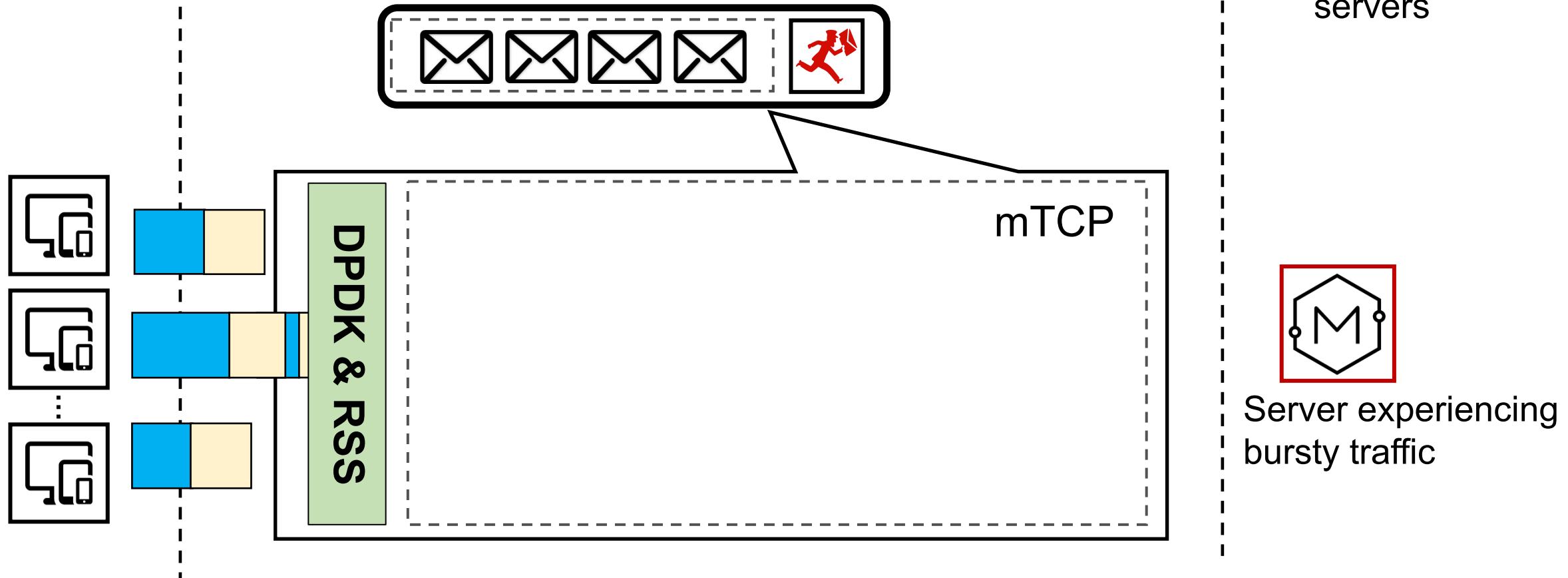


PostMan: are helpers efficient?

Clients

Helper servers

Memcached servers

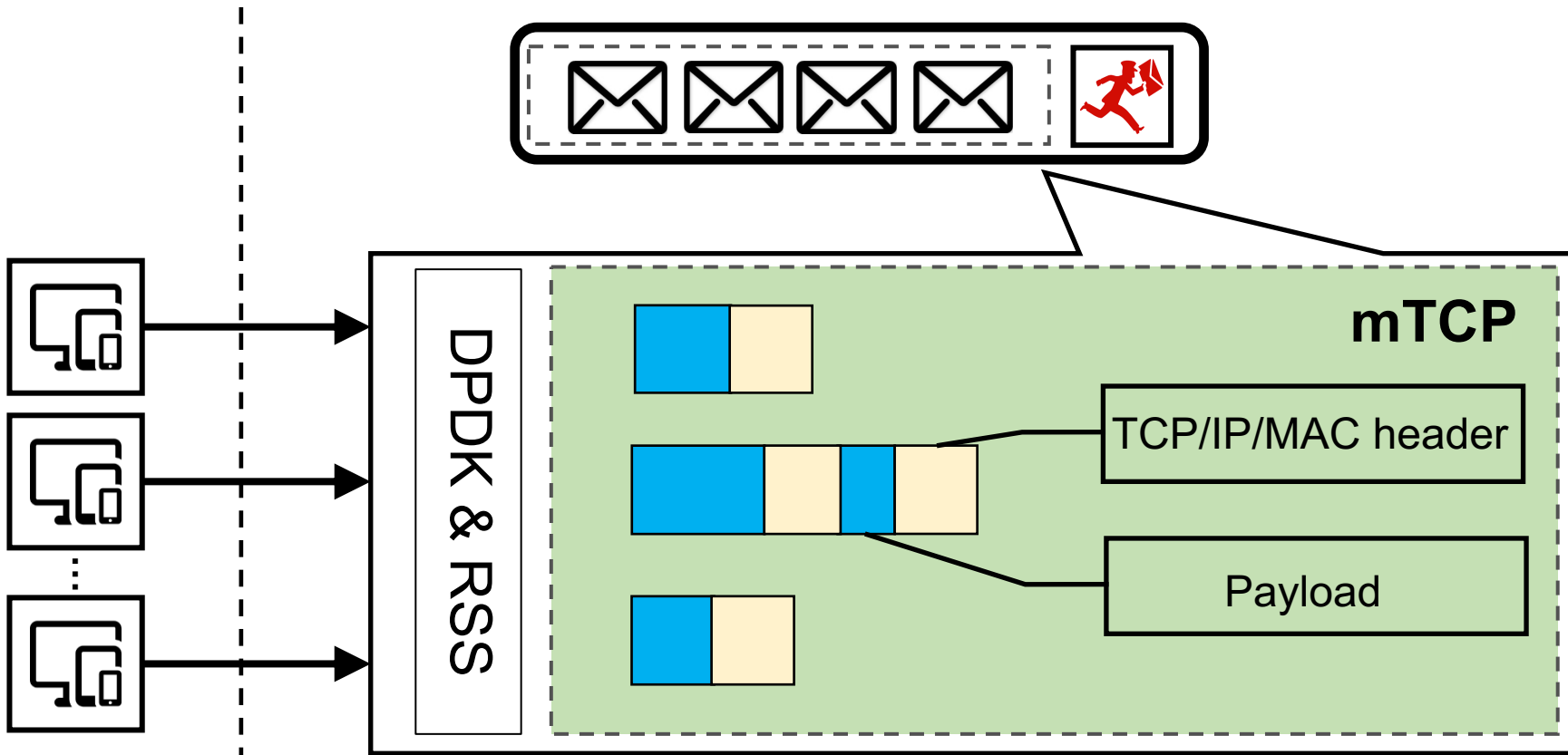


PostMan: are helpers efficient?

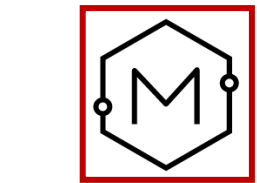
Clients

Helper servers

Memcached servers

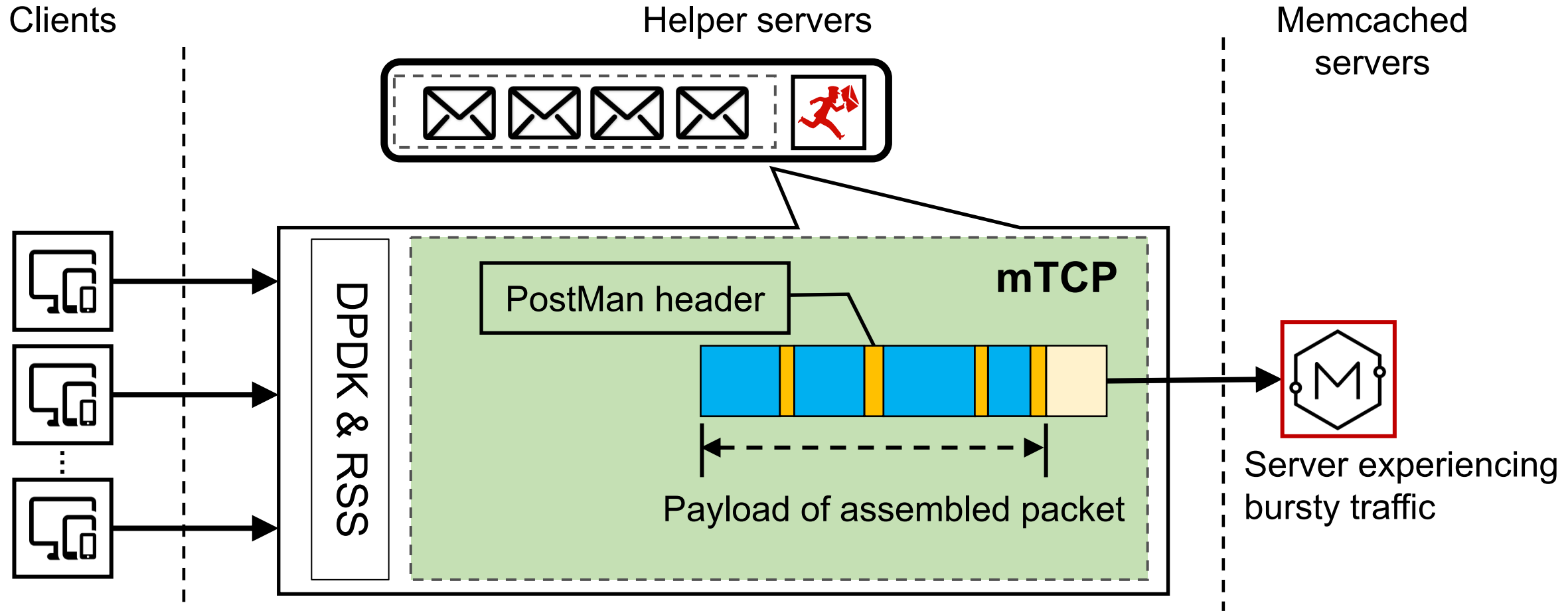


DPDK & mTCP based stack
Efficient packet processing



Server experiencing bursty traffic

PostMan: are helpers efficient?

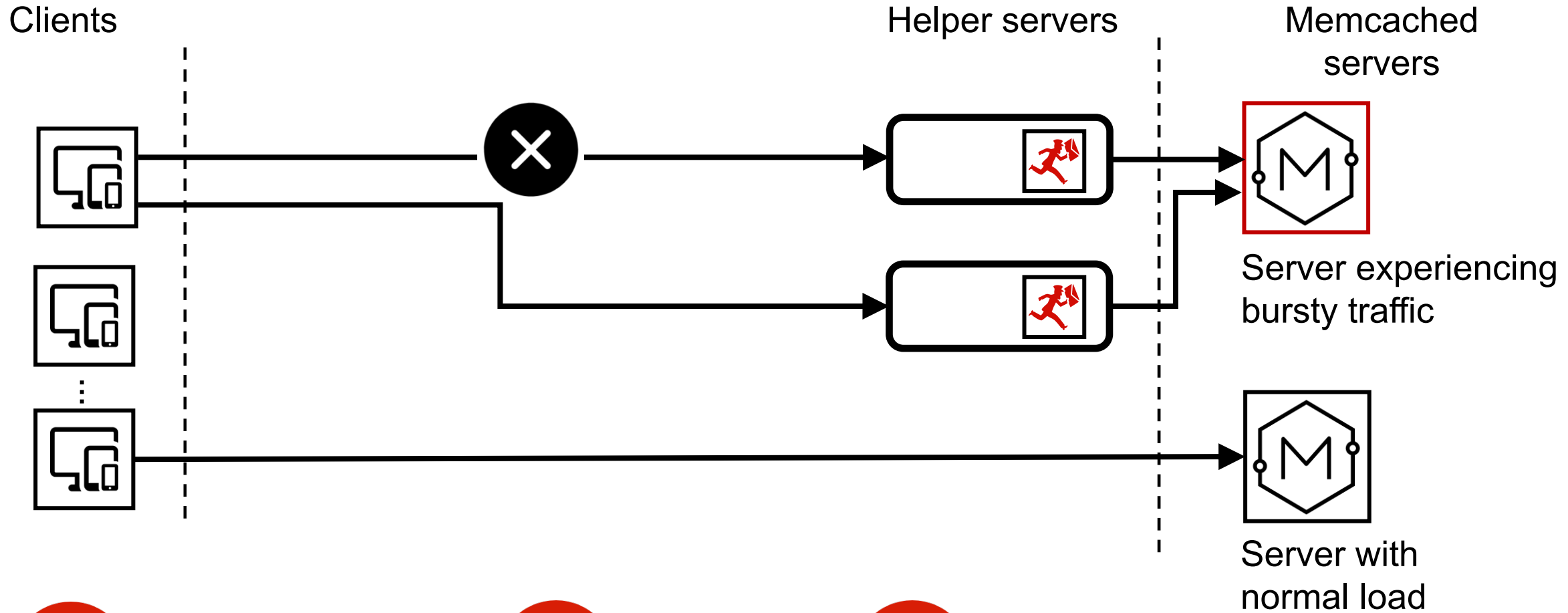


DPDK & mTCP based stack
Efficient packet processing



Remove duplicated headers
Reduce bandwidth consumption

PostMan: are helpers fault-tolerant and scalable?



Free connection migration

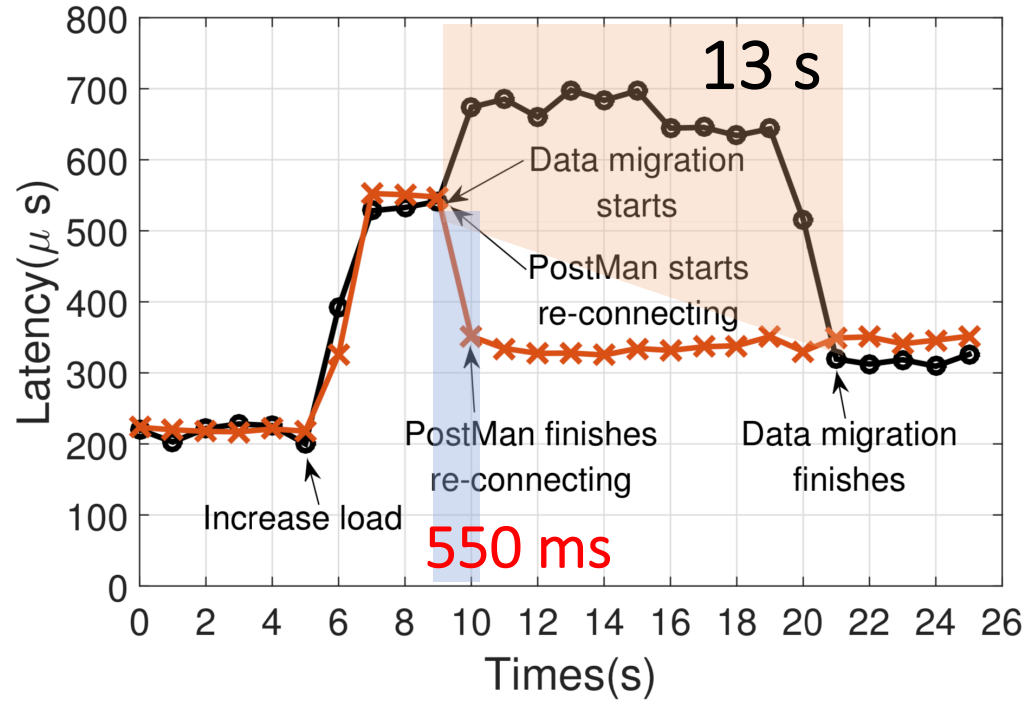


Stateless



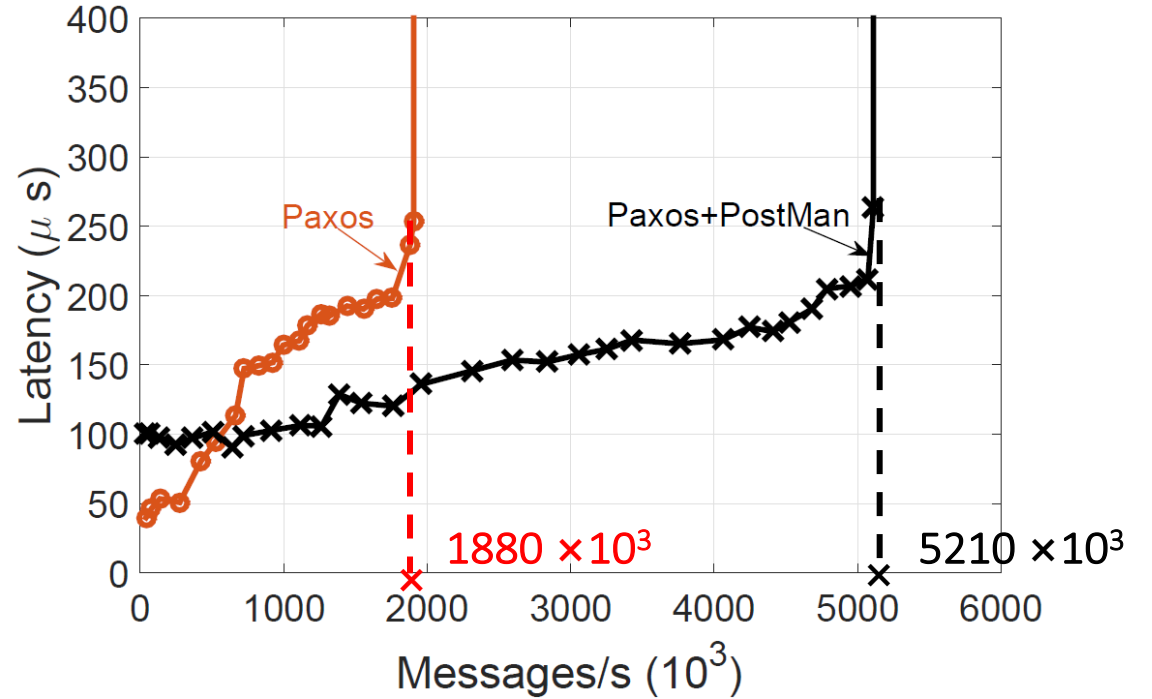
No scalability bottleneck

PostMan vs. Data migration: rapid and efficient



Mitigating bursty traffic in Memcached

Mitigation time: **550ms** vs. 13s



The latency with different load for Paxos and Paxos + PostMan

Throughput: **2.8x**

Thank you!

Track II: Networking

9:35 AM, Friday July 12, 2019