USENIX Annual Technical Conference (ATC) @ RENTON, WA, USA, JULY 12, 2019

PostMan: Rapidly Mitigating Bursty Traffic by Offloading Packet Processing

Panpan Jin[1], Jian Guo[1], Yikai Xiao[1], Rong Shi[2], Yipei Niu[1],
Fangming Liu*[1], Chen Qian[3], Yang Wang[2]

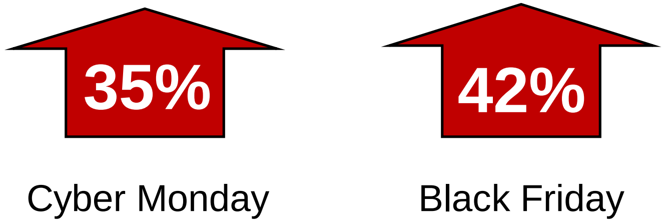
[1]OpenCloudNeXt Group, Huazhong University of Science and Technology
[2]The Ohio State University
[3]University of California Santa Cruz

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

Peak shopping season is going global

- Sale on Cyber Monday hits new record at $6.6 Billion in 2017[1]
- Black Friday racks up $5.03 billion in online sales in 2017[2]
- The 24-hour sale on Nov. 11 reaches $25 billion in sales in 2017[3]

- Bursty traffic is arriving!
  - Conversation rate in 24h[4]
  - Statistics on double 11[5]
    - ✓ **325,000** orders/s at peak
    - ✓ **256,000** transactions/s at peak

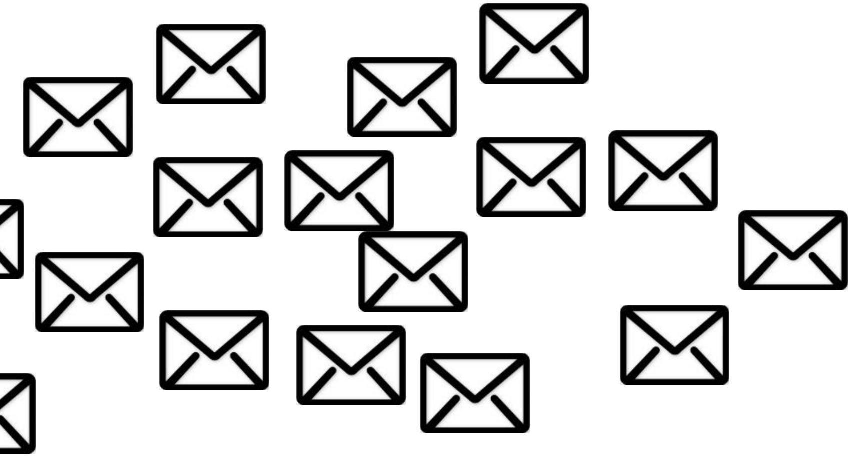

35%

Cyber Monday

42%

Black Friday

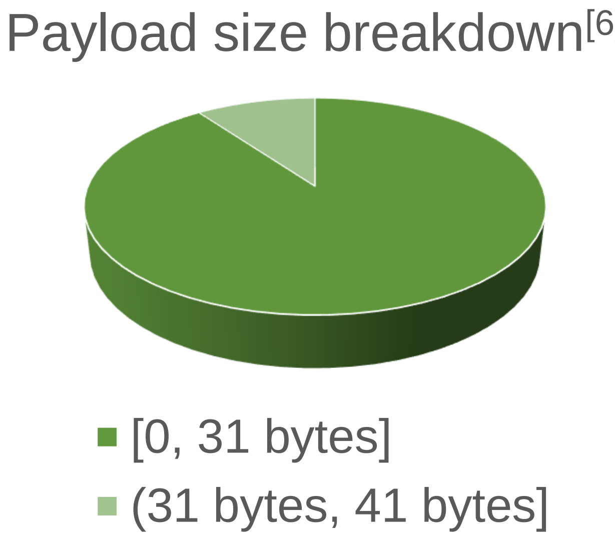
Bursty traffic is a headache!

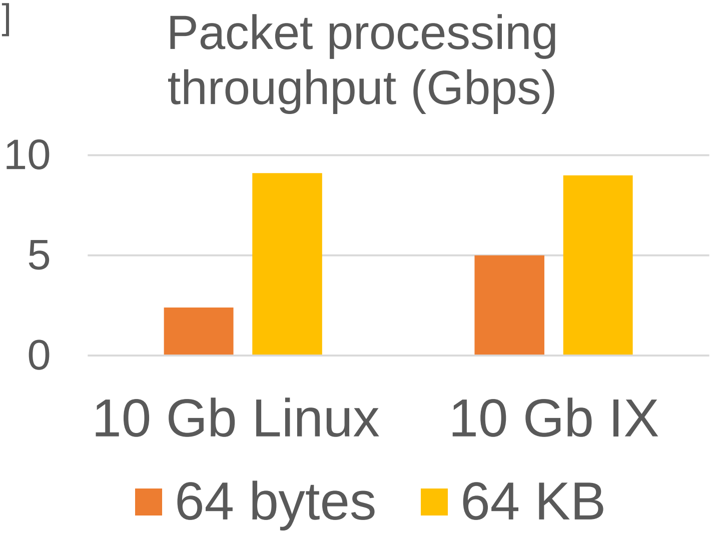


Servers

🚨 Large volume

🚨 Short duration

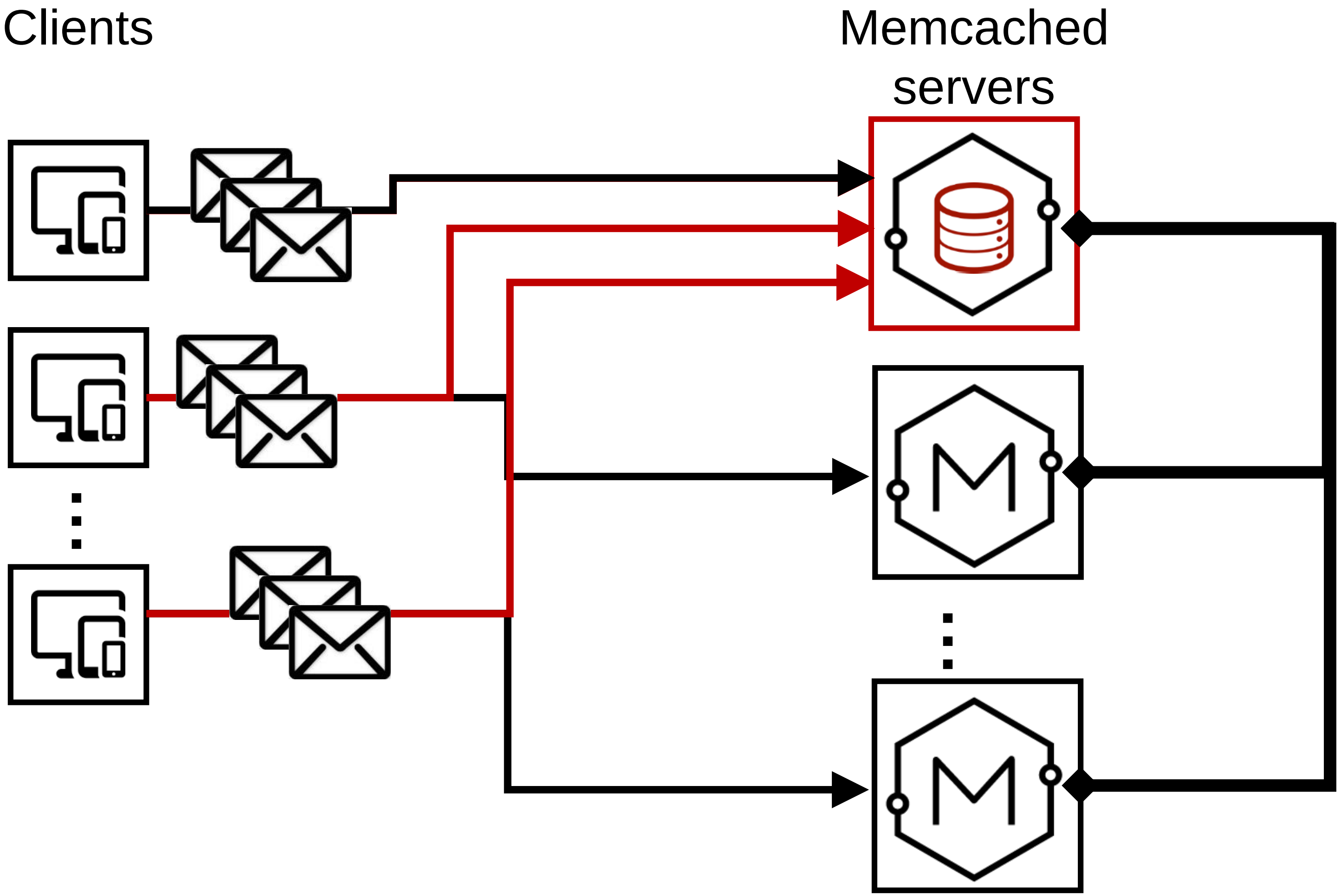




☹ Small packets

☹ Severe overhead

Traditional remedy: migrating hot data for load balancing

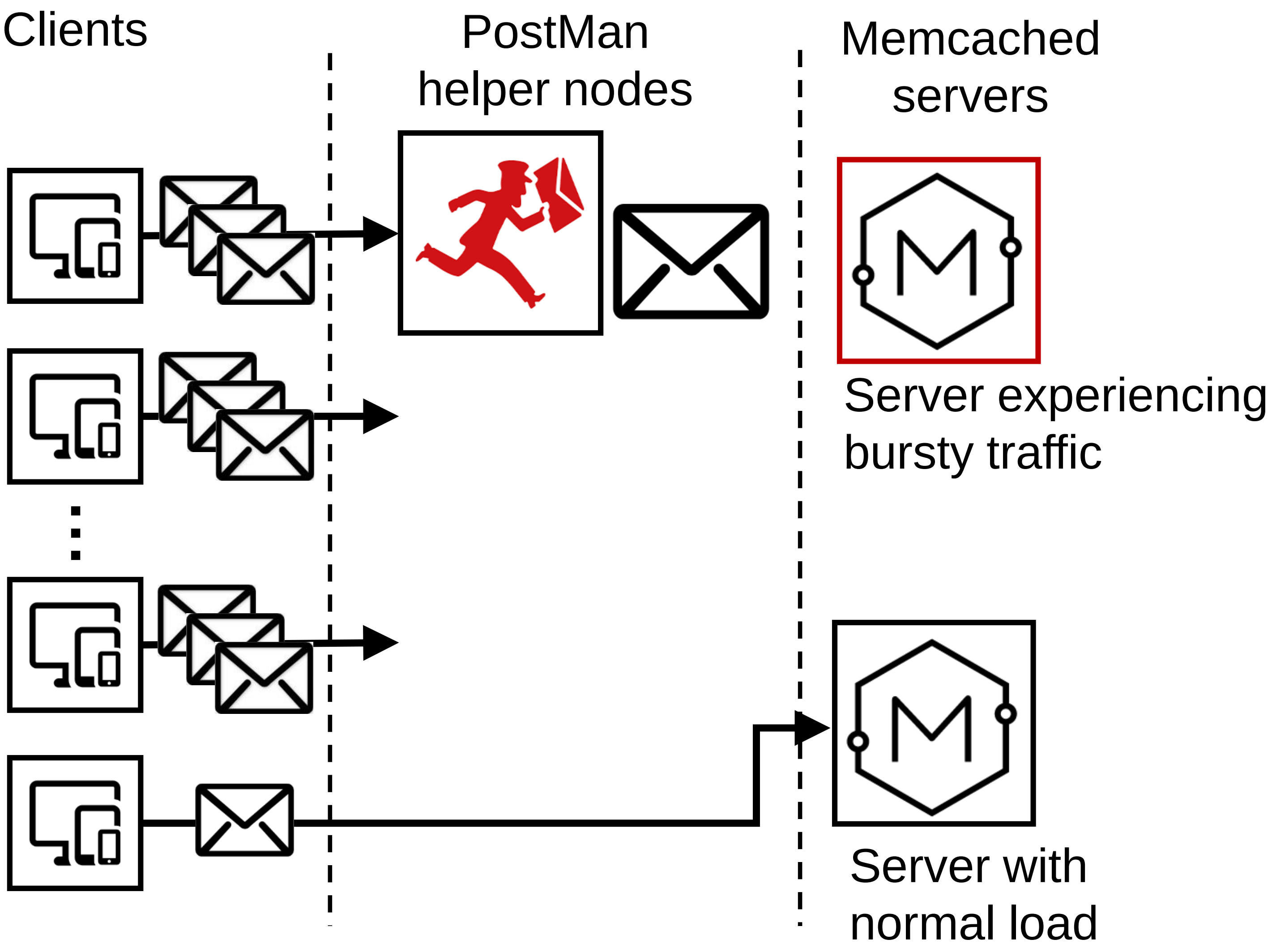


😞 Time-consuming

😞 Exacerbate overload

😞 High packet processing overhead

PostMan: batching and offloading on demand



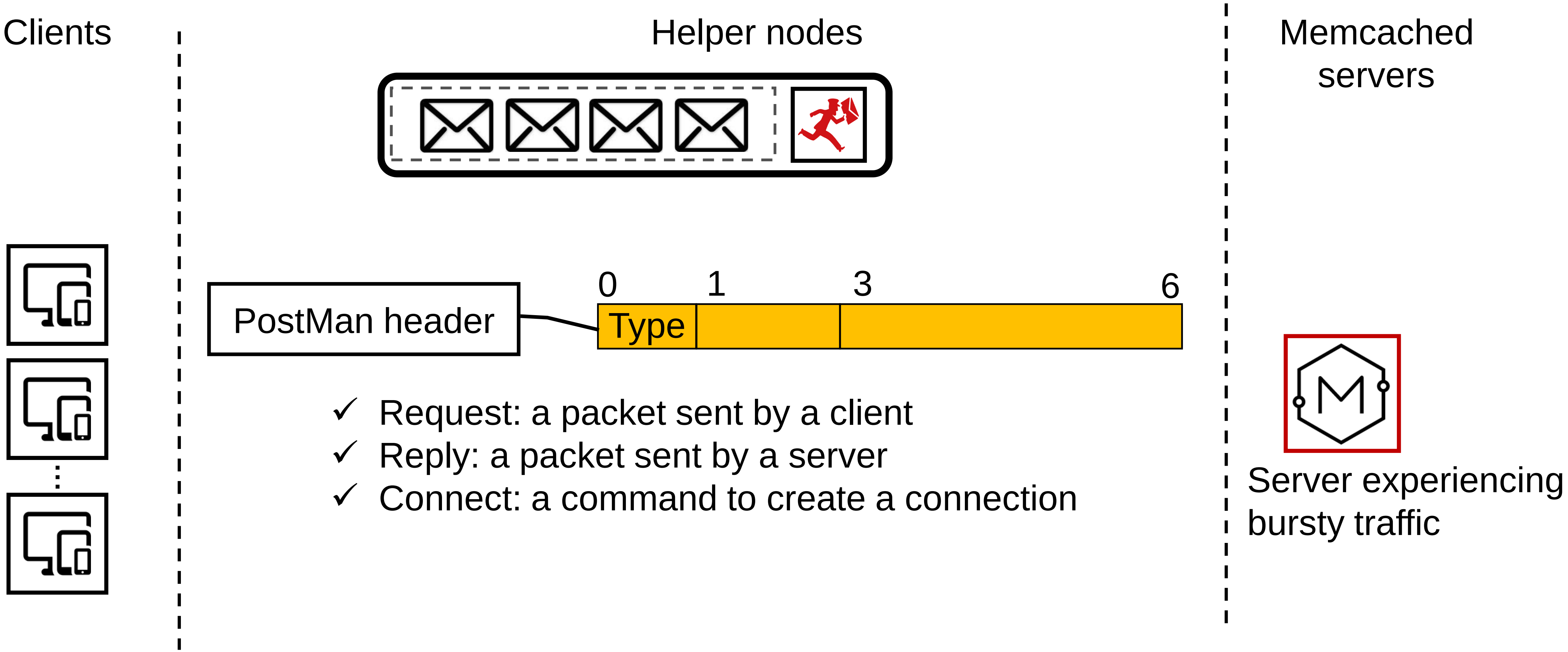
- Helper batches small packets into large ones
- PostMan offloads packet overhead from overloaded server to helpers

☺ Large packets

☺ No data migration

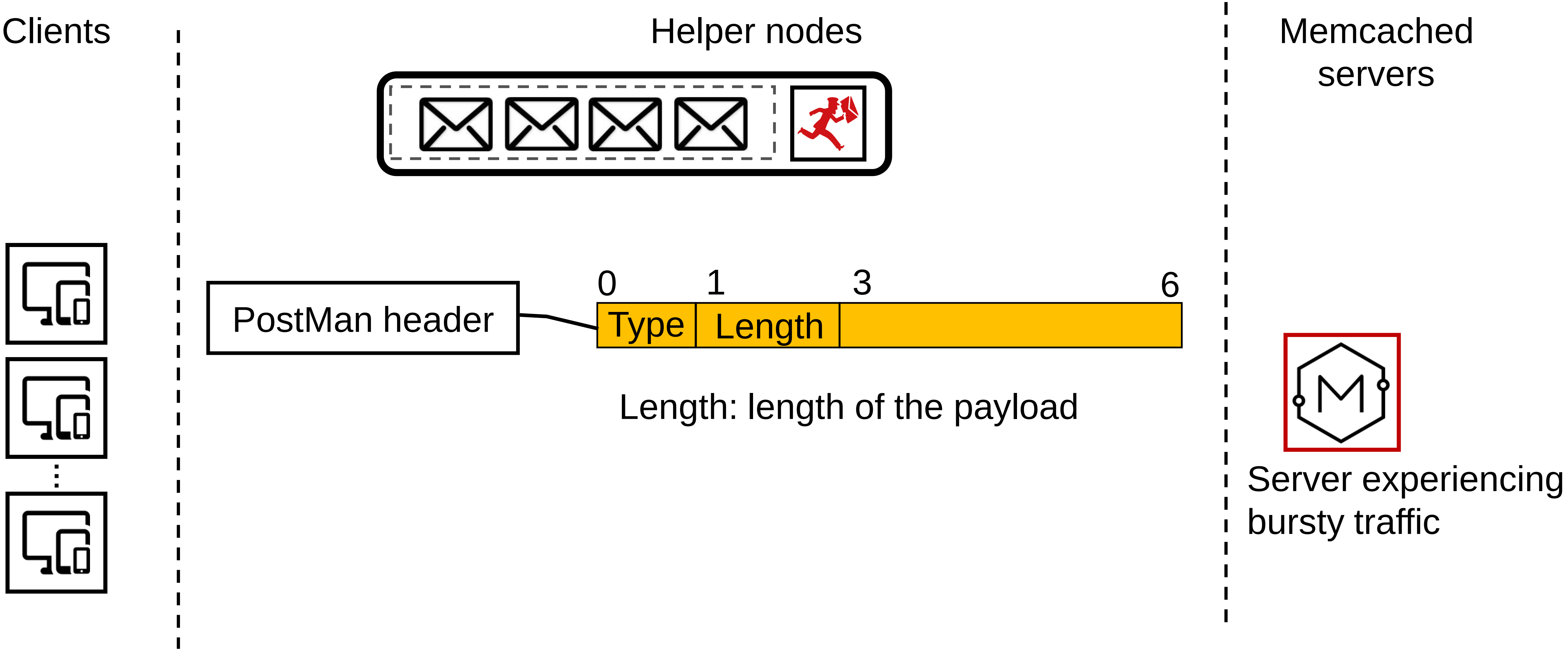
☺ Rapid mitigation

How to assemble small packets?



- ✓ Request: a packet sent by a client
- ✓ Reply: a packet sent by a server
- ✓ Connect: a command to create a connection

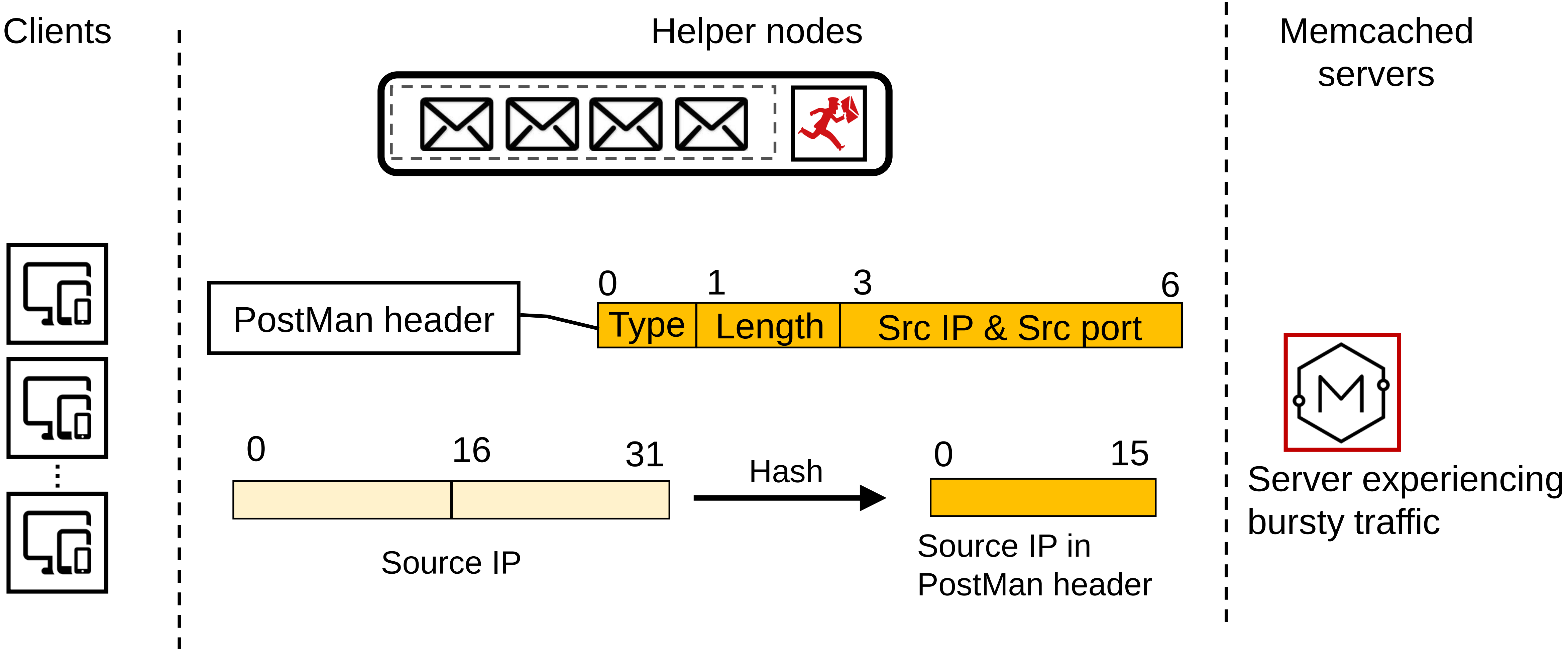
How to assemble small packets?



Length: length of the payload

Server experiencing bursty traffic

How to assemble small packets?



Clients

Helper nodes

Memcached servers

PostMan header

0 1 3 6

Type | Length | Src IP & Src port

0 16 31

Source IP

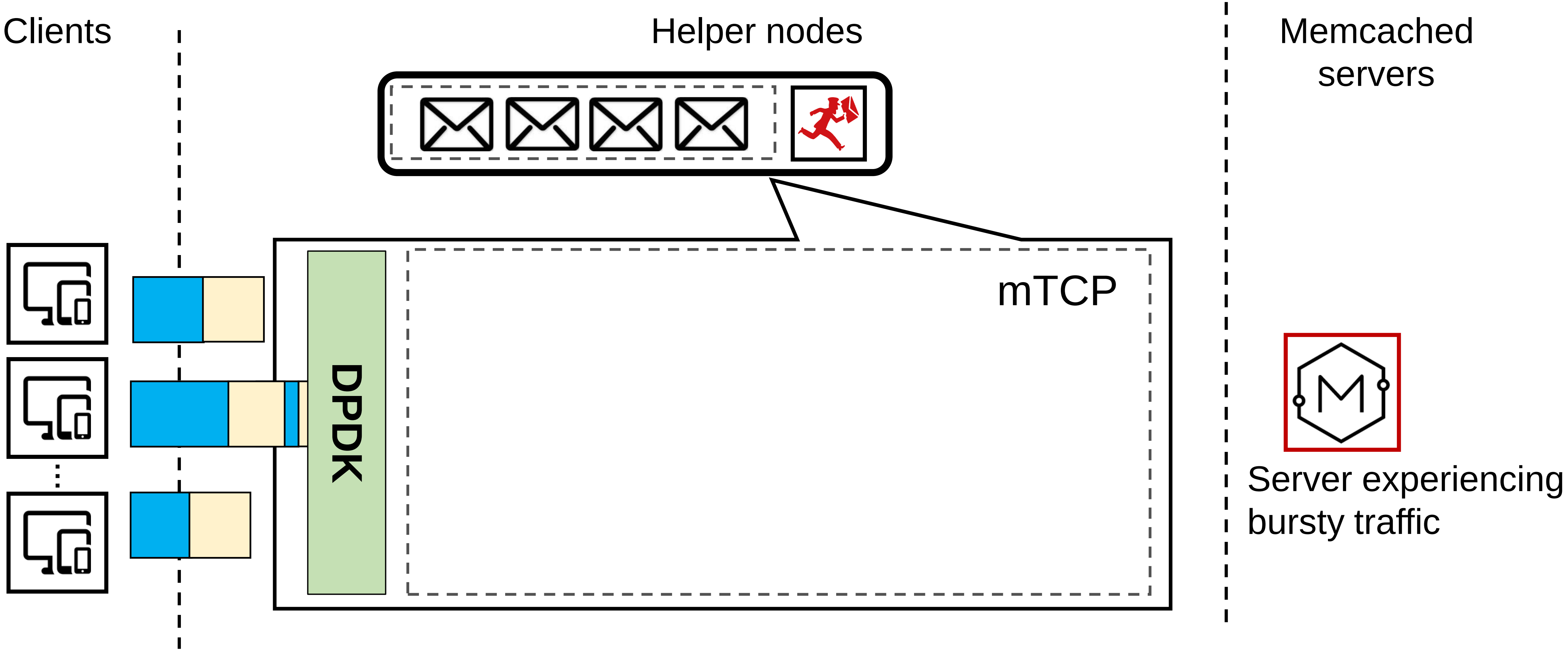
Hash

0 15

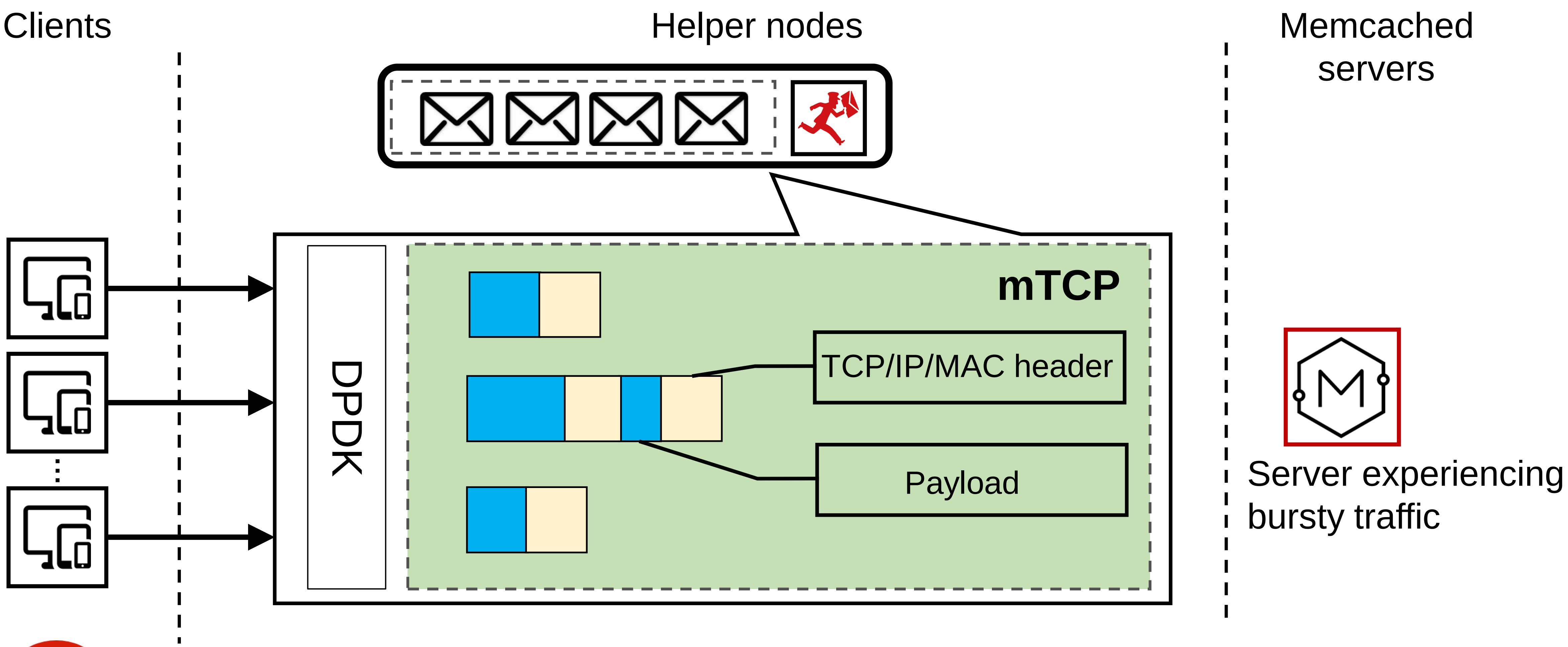
Source IP in PostMan header

Server experiencing bursty traffic

Is batching in helpers efficient?

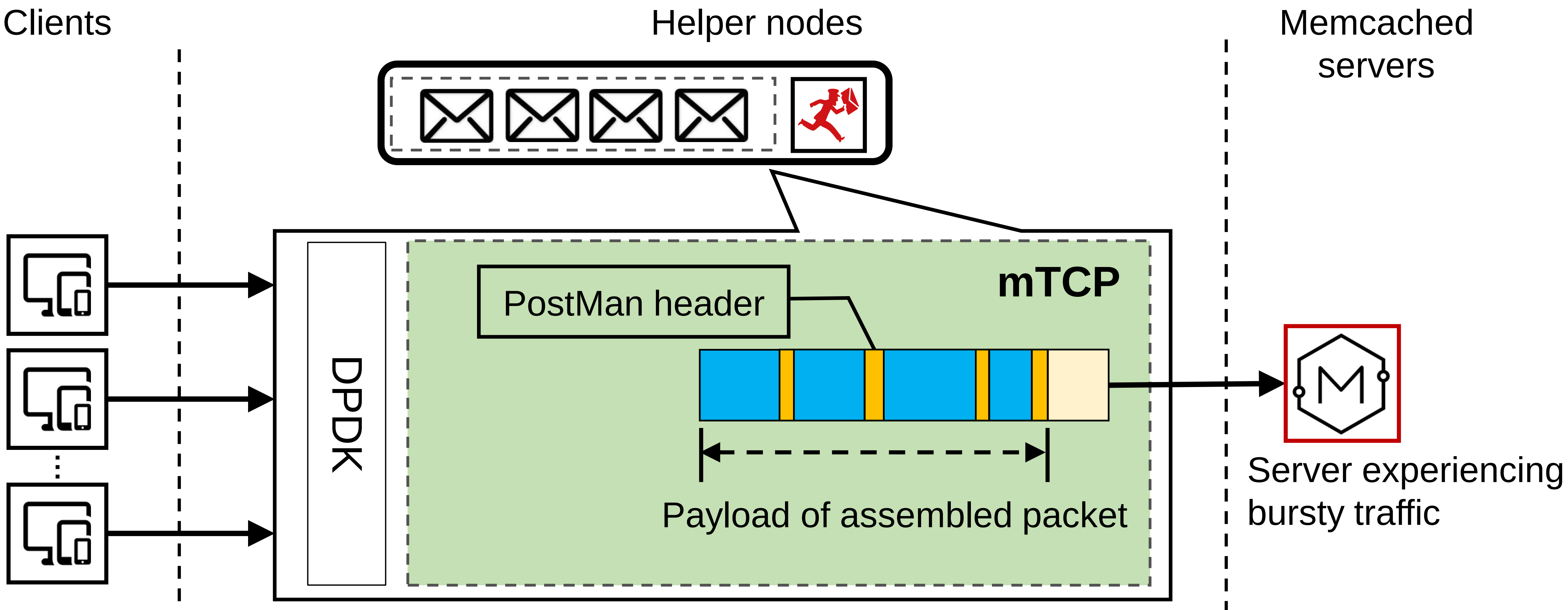


Is batching in helpers efficient?



DPDK & mTCP based stack
Efficient packet processing

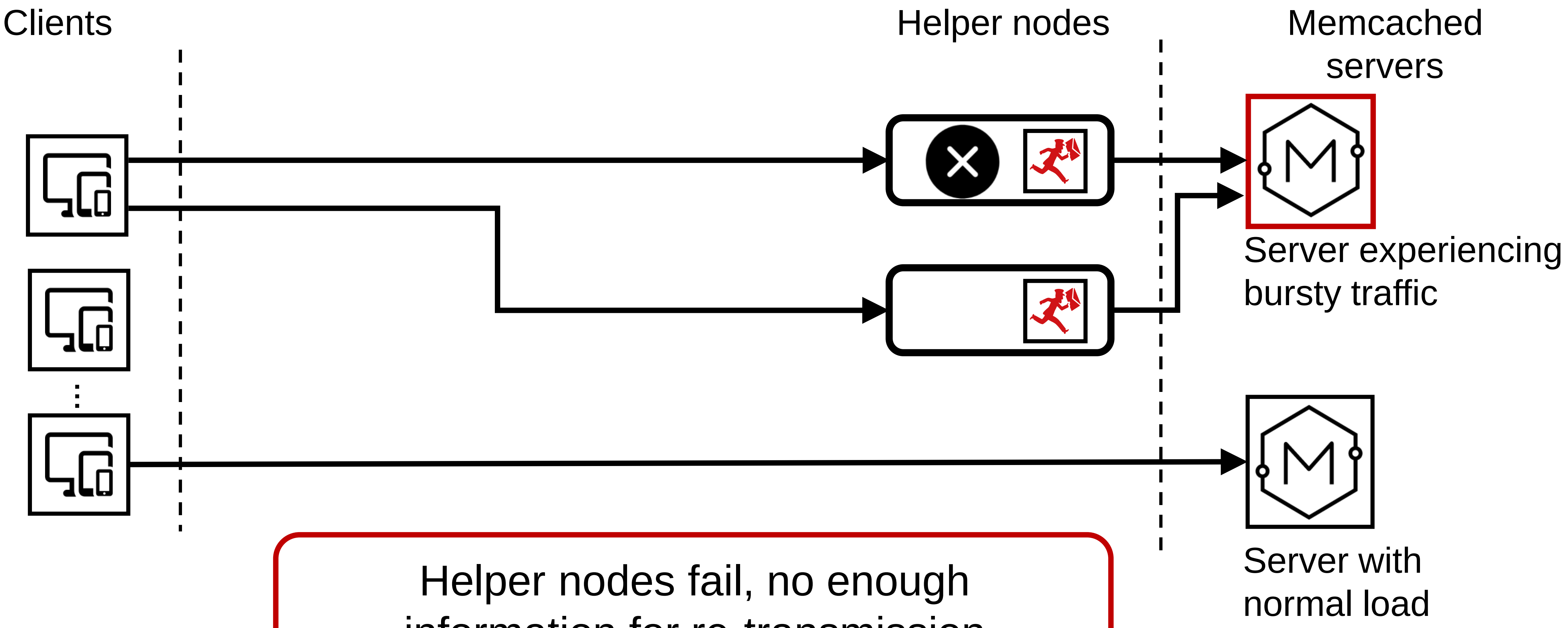
Is batching in helpers efficient?


Clients
Helper nodes
Memcached servers
DPDK
mTCP
PostMan header
Payload of assembled packet
Server experiencing bursty traffic

✓ DPDK & mTCP based stack
Efficient packet processing

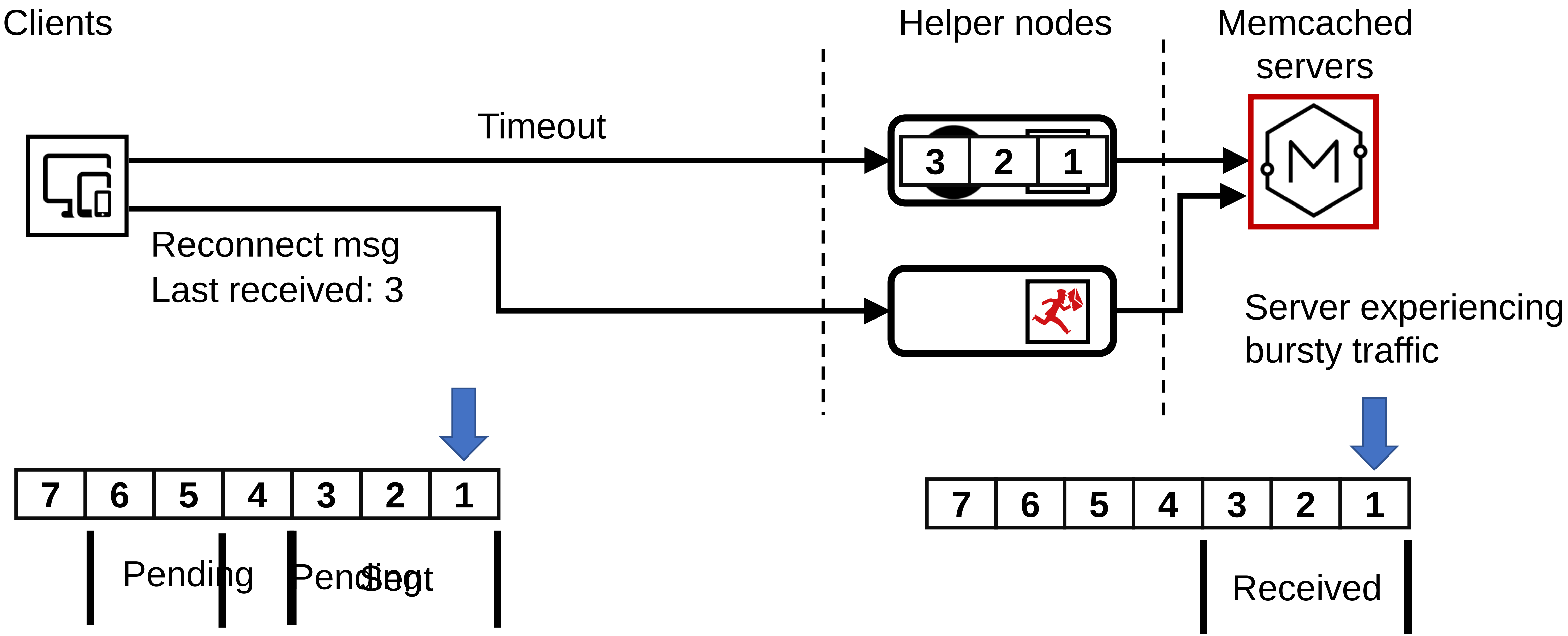
✓ Remove duplicated headers
Alleviate packet processing overhead

Everything works fine, except…



Helper nodes fail, no enough information for re-transmission

Stateless failover mechanism

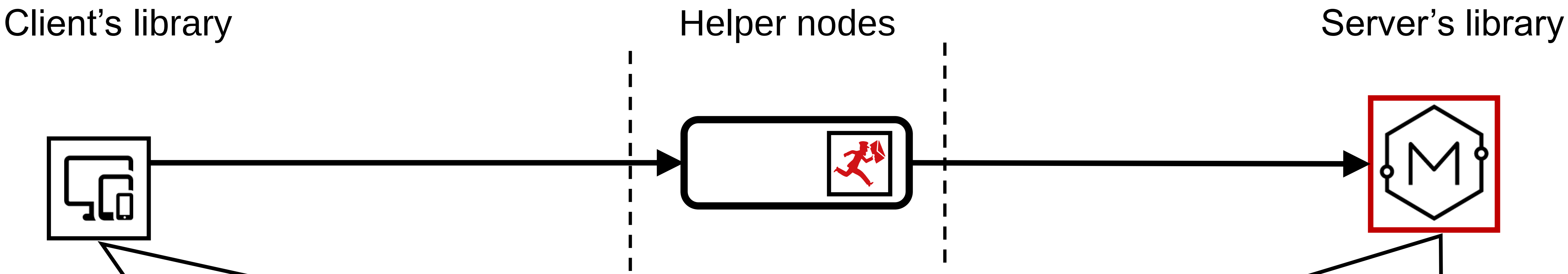


✓ Freely migrating connection

✓ Stateless

✓ No scalability bottleneck

Programming with PostMan library



pm_connect:

- ✓ Chooses a helper and connect to the helper.
- ✓ Sends a special “connect” packet to the helper node.

get_info:

- ✓ Allows the application to retrieve connection information, such as the number of sent and received packets

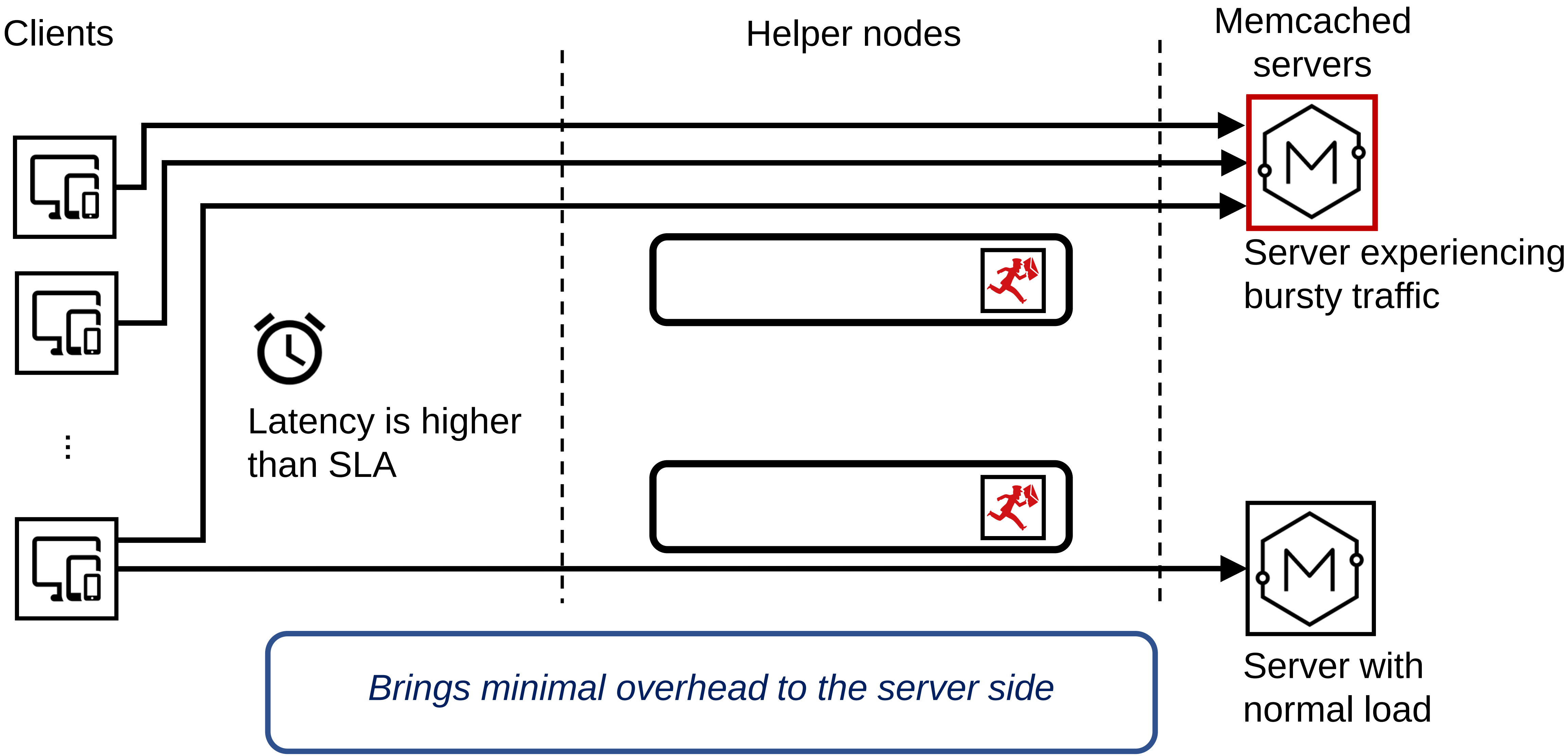
decompose:

- ✓ Identifies the “connect” packet and notifies application that a new client tries to connect
- ✓ Disassembles the packet into small packets

compose:

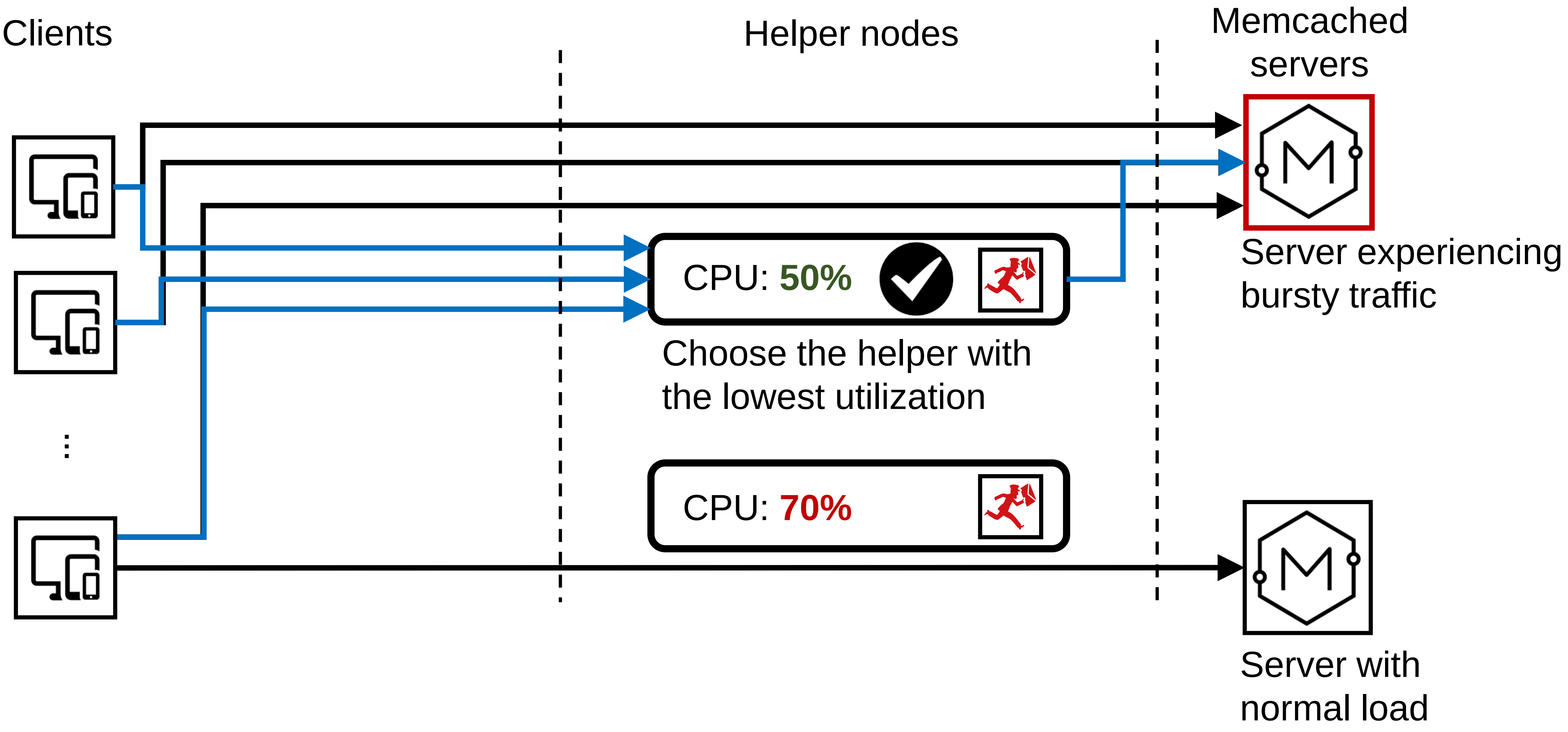
- ✓ Buffers multiple replies and assemble them

How to enable helpers?


Clients
Helper nodes
Memcached servers
Server experiencing bursty traffic
Latency is higher than SLA
Server with normal load

Brings minimal overhead to the server side

Load balancing across helpers



Clients

Helper nodes

Memcached servers

CPU: 50%

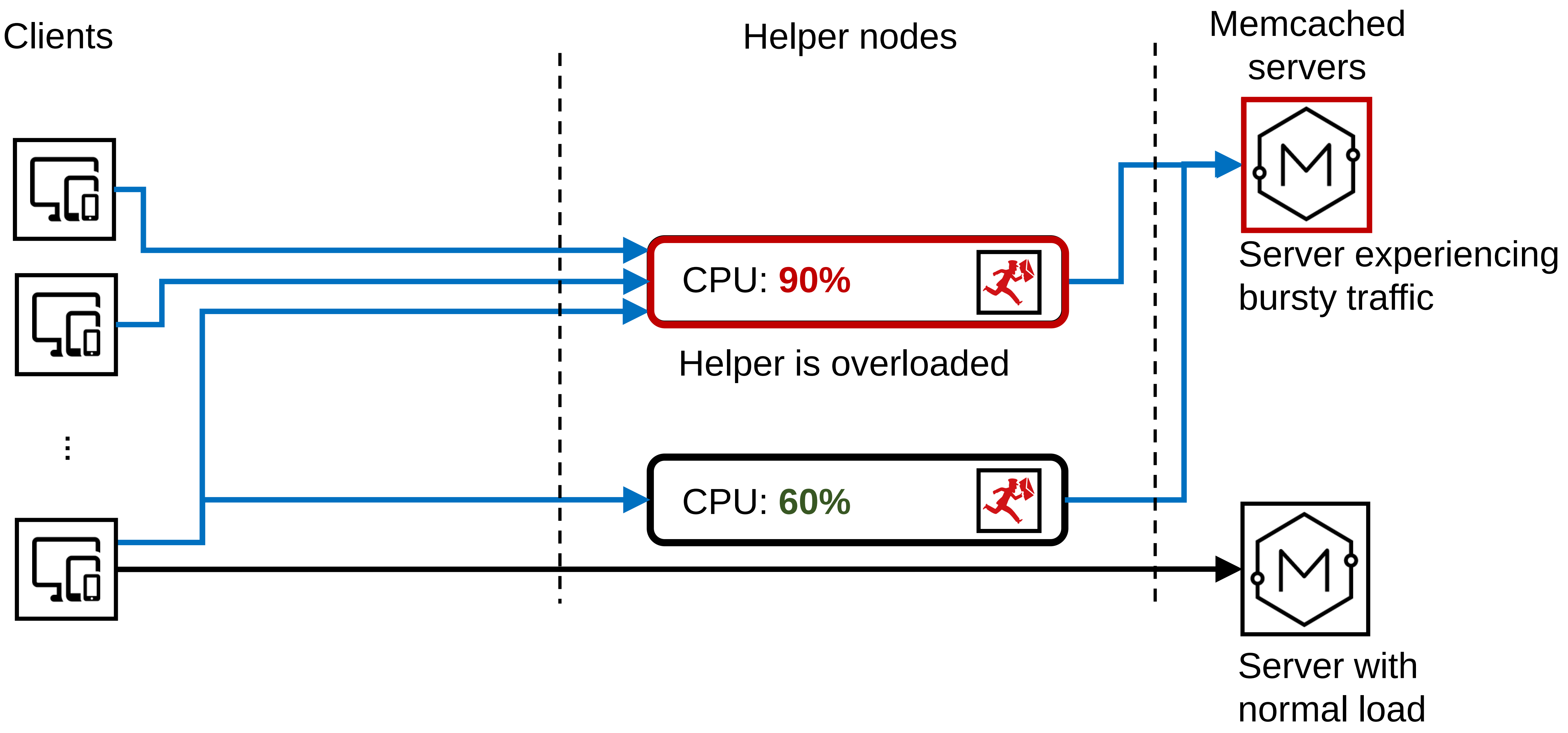
Choose the helper with the lowest utilization

CPU: 70%

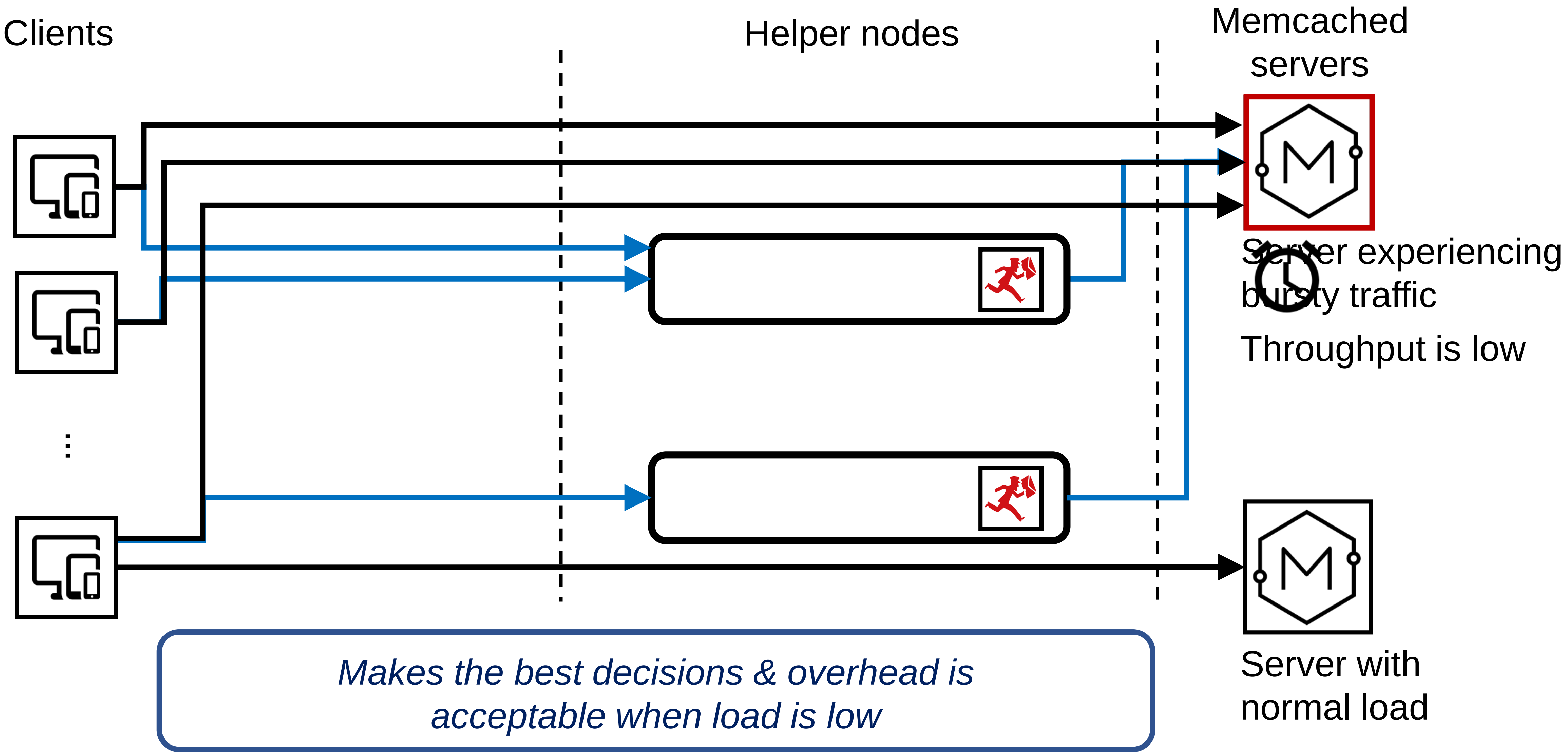
Server experiencing bursty traffic

Server with normal load

Load balancing across helpers

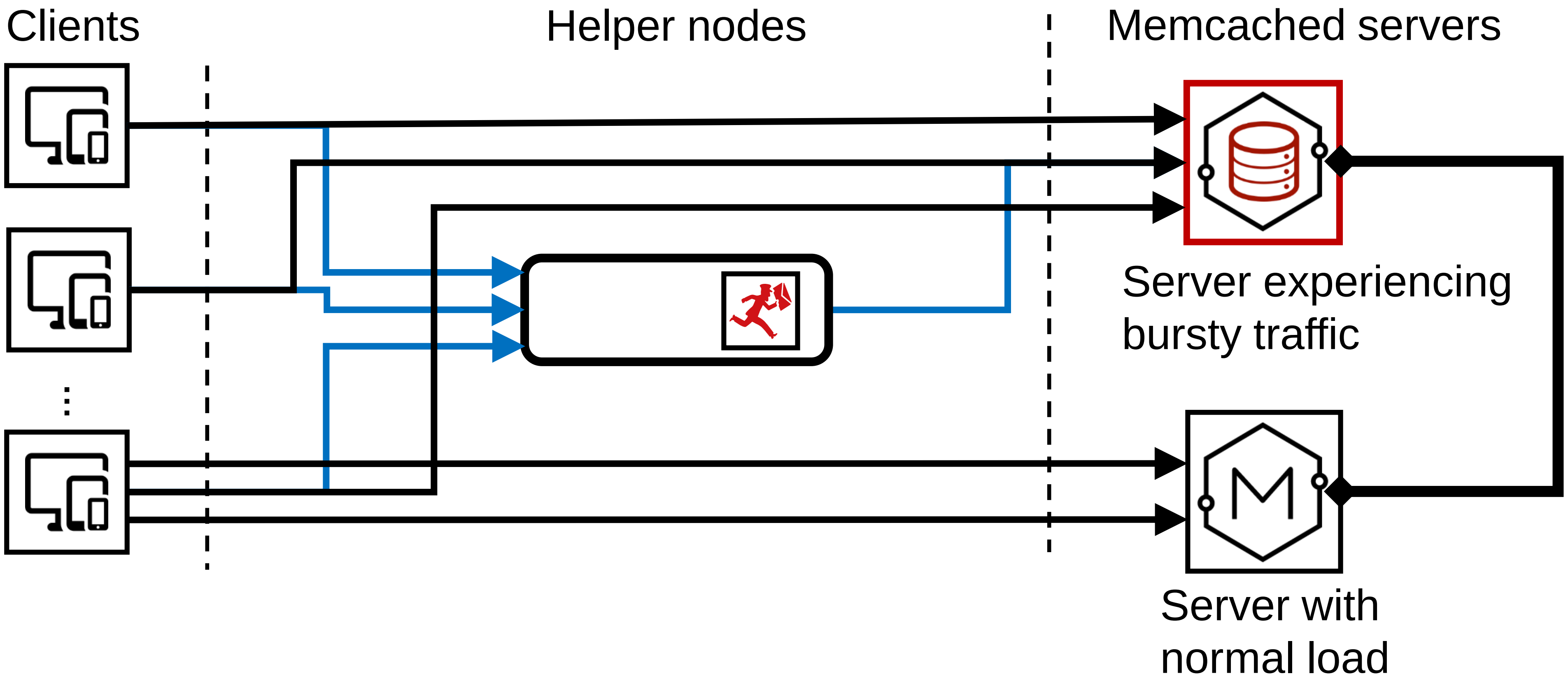


How to disable helpers?



Makes the best decisions & overhead is acceptable when load is low

Complementary to data migration



- Positioning of PostMan
  - PostMan is an alternative solution to data migration for bursty traffic
  - Data migration is the ultimate solution to mitigate bursty traffic

Evaluation Setup

□ Testbed: CloudLab: 15 machines

□ Machine: 10 physical cores and hyper-threading, an Intel 82599ES 10 Gigabit NIC

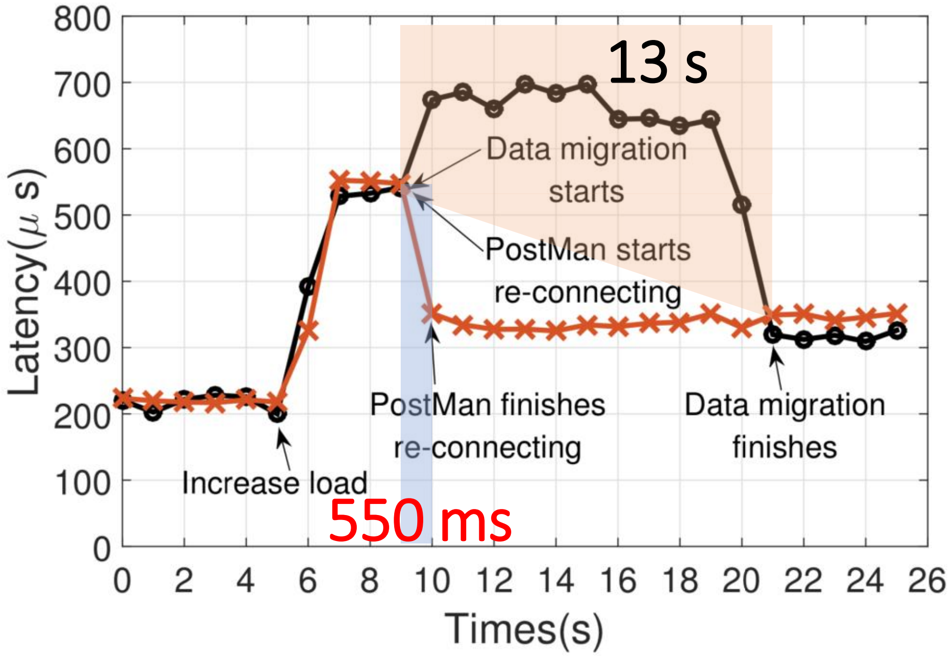
□ Server side: Memcached , Paxos and IX

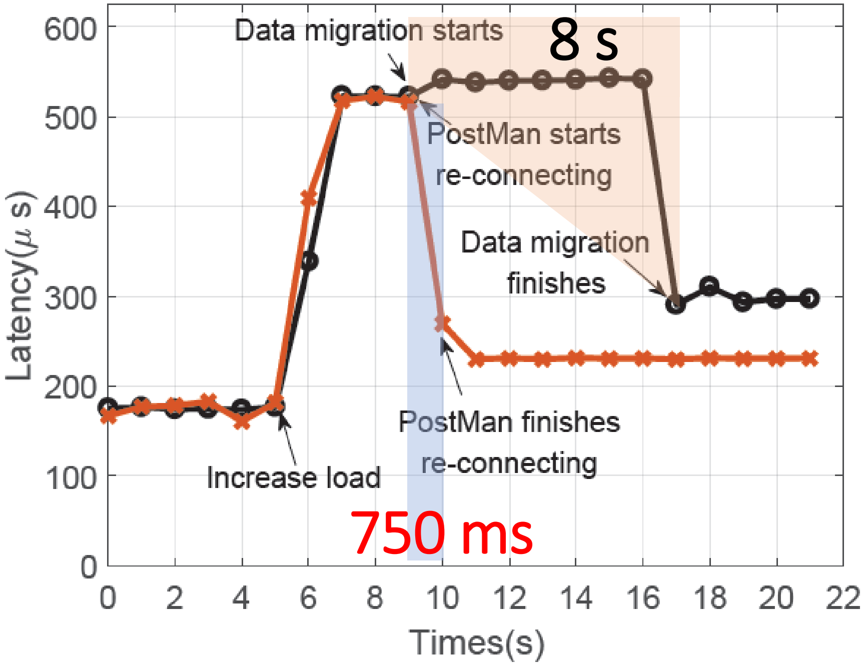
□ Helper node: DPDK 16.07.2

□ Client side: Ping-pong benchmark and IX

□ SLA: 500 μs (99 percentile latency, p99)

PostMan vs. Data Migration

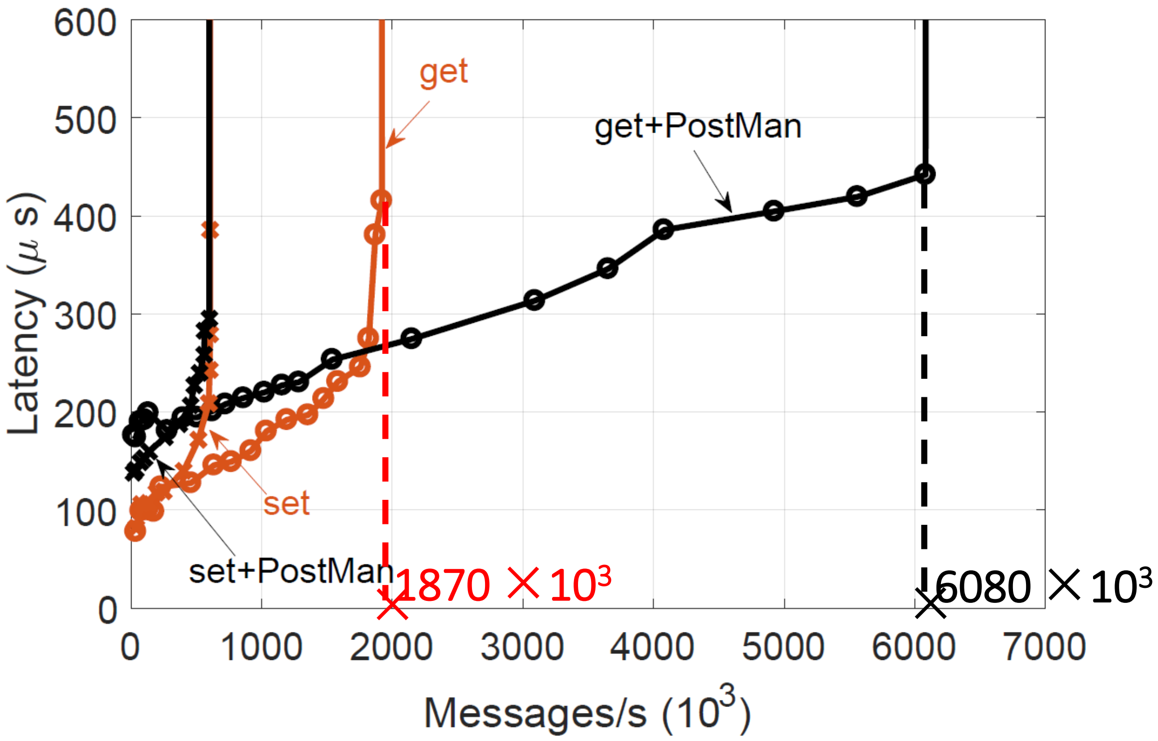




2 helper nodes
660 client connections
Mitigation time: **550ms** vs. 13s

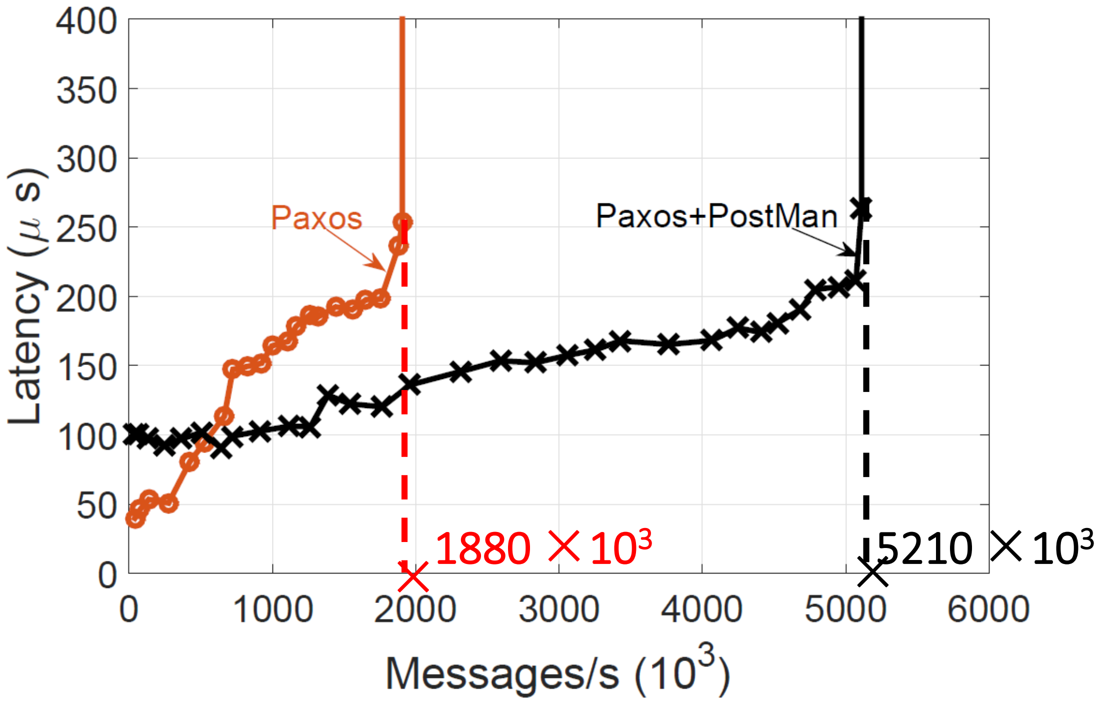
2 helper nodes
960 client connections
Mitigation time: **750ms** vs. 8s

Capabilities of PostMan



Memcached vs Memcached + PostMan

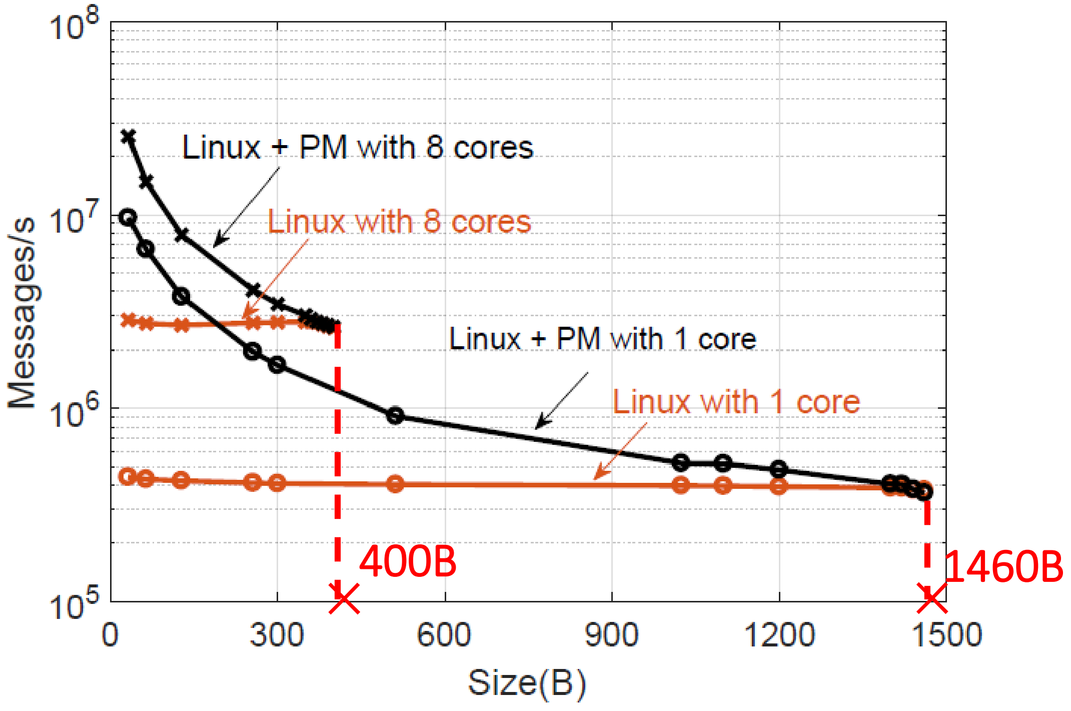
- Up to 5 helper nodes
- Load range: **2000K~6000K**
- Throughput: **3.3**×

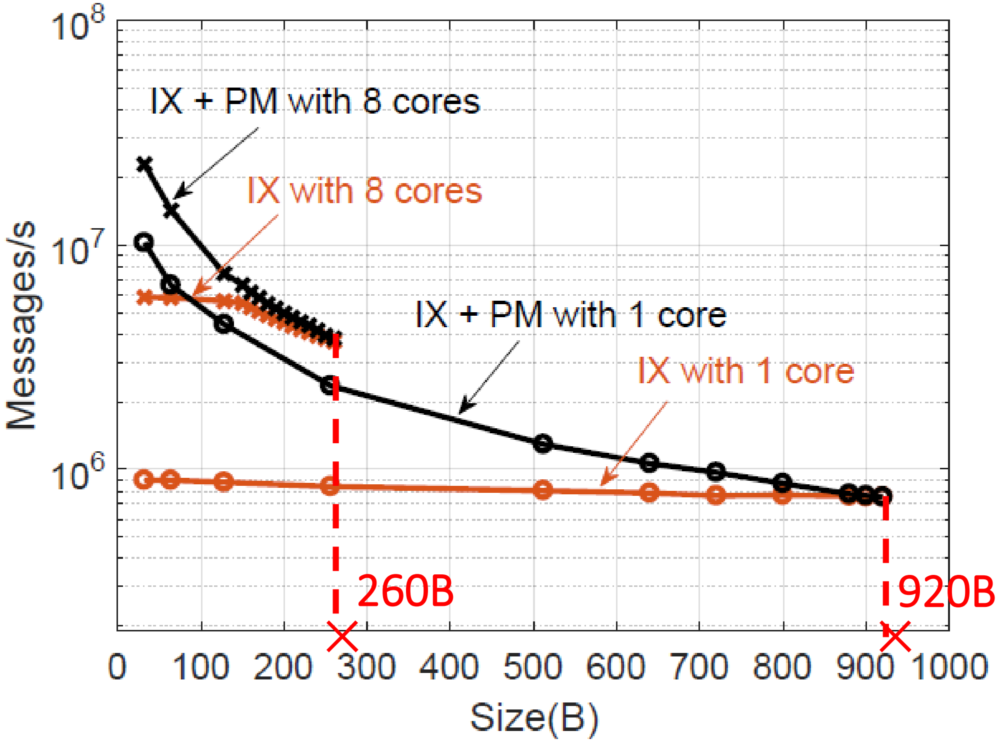


Paxos vs Paxos + PostMan

- Up to 6 helper nodes
- Load range: **500K~5000K**
- Throughput: **2.8**×

Performance gain of PostMan





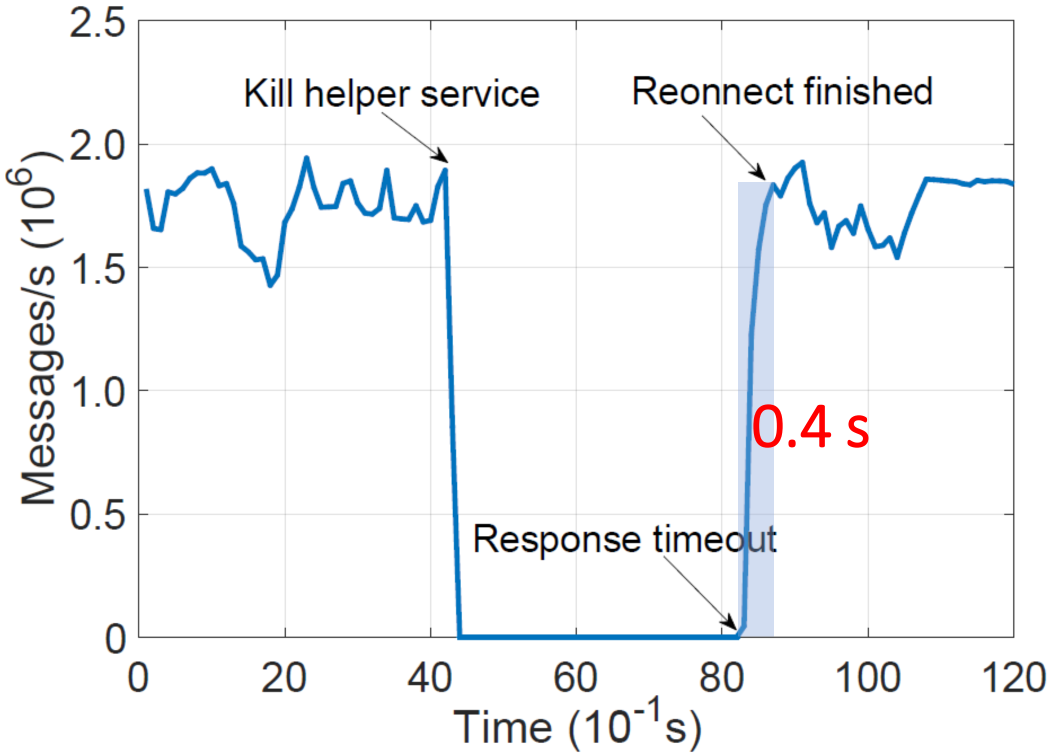
Linux vs Linux + PostMan

- Up to 6 helper nodes
- 8 cores: turning point is **400 bytes**
- 1 core: turning point is **1460 bytes**

IX vs IX + PostMan

- Up to 6 helper nodes
- 8 cores: turning point is **260 bytes**
- 1 core: turning point is **920 bytes**

Fault tolerance



Takes about **0.4s** to recover all 1000 connections

Conclusion

We propose **PostMan**, an alternative approach to *rapidly* mitigate load imbalance for services processing *small requests*

- ✓ Rapid: much faster than data migration
- ✓ Efficient: Fast I/O, user-level stack for packet processing
- ✓ Fault-tolerant: stateless failover design
- ✓ Scalable: no scalability bottleneck

Reference

[1] Cyber Monday Hits New Record At $6.6 Billion, The Largest Online Shopping Day In U.S. History, https://www.forbes.com/sites/jeanbaptiste/2017/11/28/report-cyber-monday-hits-new-record-at-6-6-billion-over-1-billion-more-than-2016/#124347253662

[2] Black Friday racks up $5.03B in online sales, $2B on mobile alone, https://techcrunch.com/2017/11/24/black-friday-deals-net-640m-in-sales-so-far-mobile-60-of-all-traffic/

[3] Alibaba's Singles' Day By The Numbers: A Record $25 Billion Haul, https://www.forbes.com/sites/helenwang/2017/11/12/alibabas-singles-day-by-the-numbers-a-record-25-billion-haul/#4af0c1b1db15

[4] STATE OF ONLINE RETAIL PERFORMANCE, https://www.akamai.com/us/en/multimedia/documents/report/akamai-state-of-online-retail-performance-2017-holiday.pdf

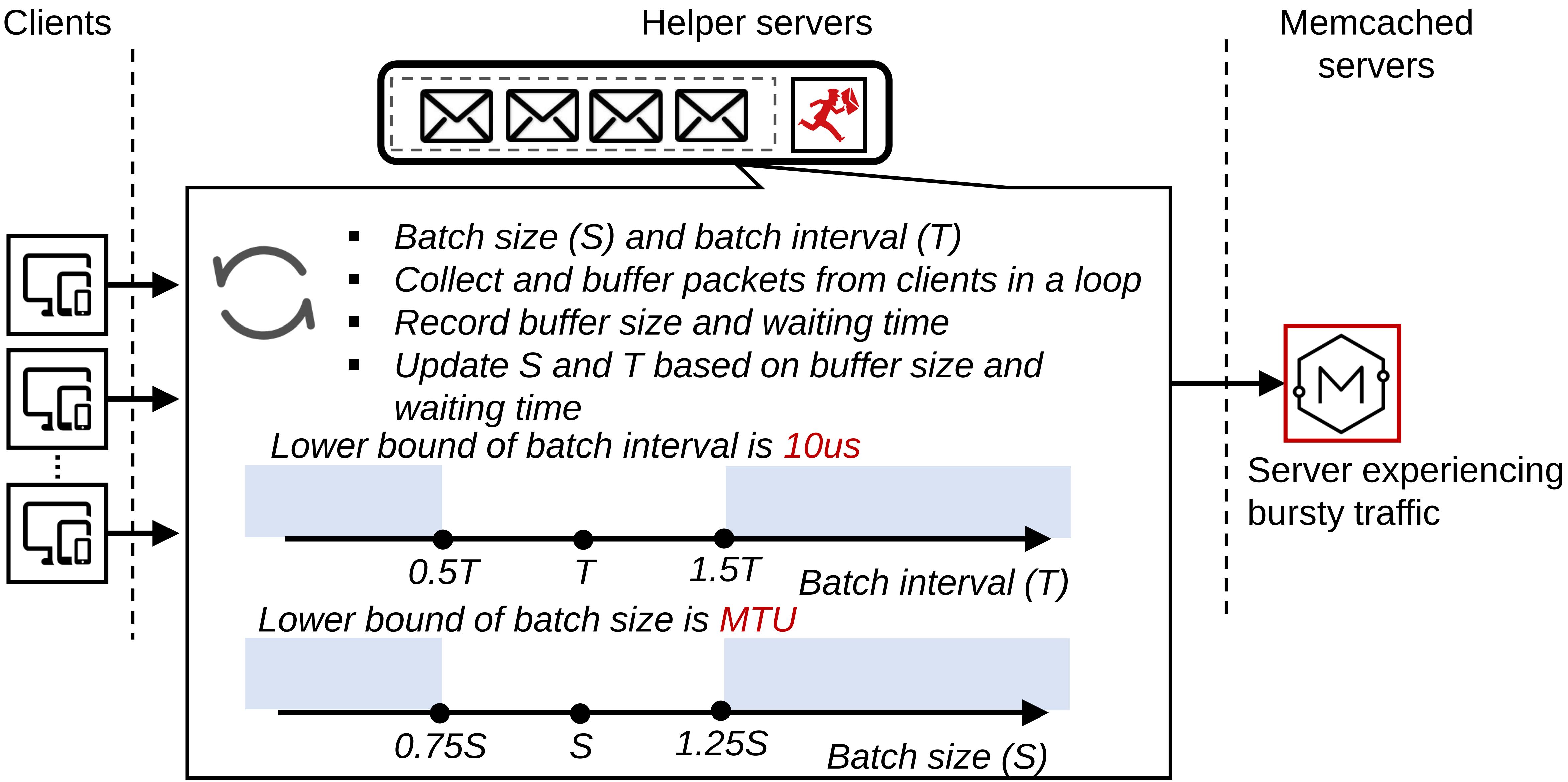
[5] $25 billion in 24 hours: Alibaba creates history. Highlights from Double 11 at Shanghai, https://yourstory.com/2017/11/25-billion-24-hours-alibaba-creates-history-double-11

[6] Workload Analysis of a Largescale Key-value Store. In Proc. of SIGMETRICS, 2012.

Thank you!

Scan for more!

Is batching in helpers adaptive?



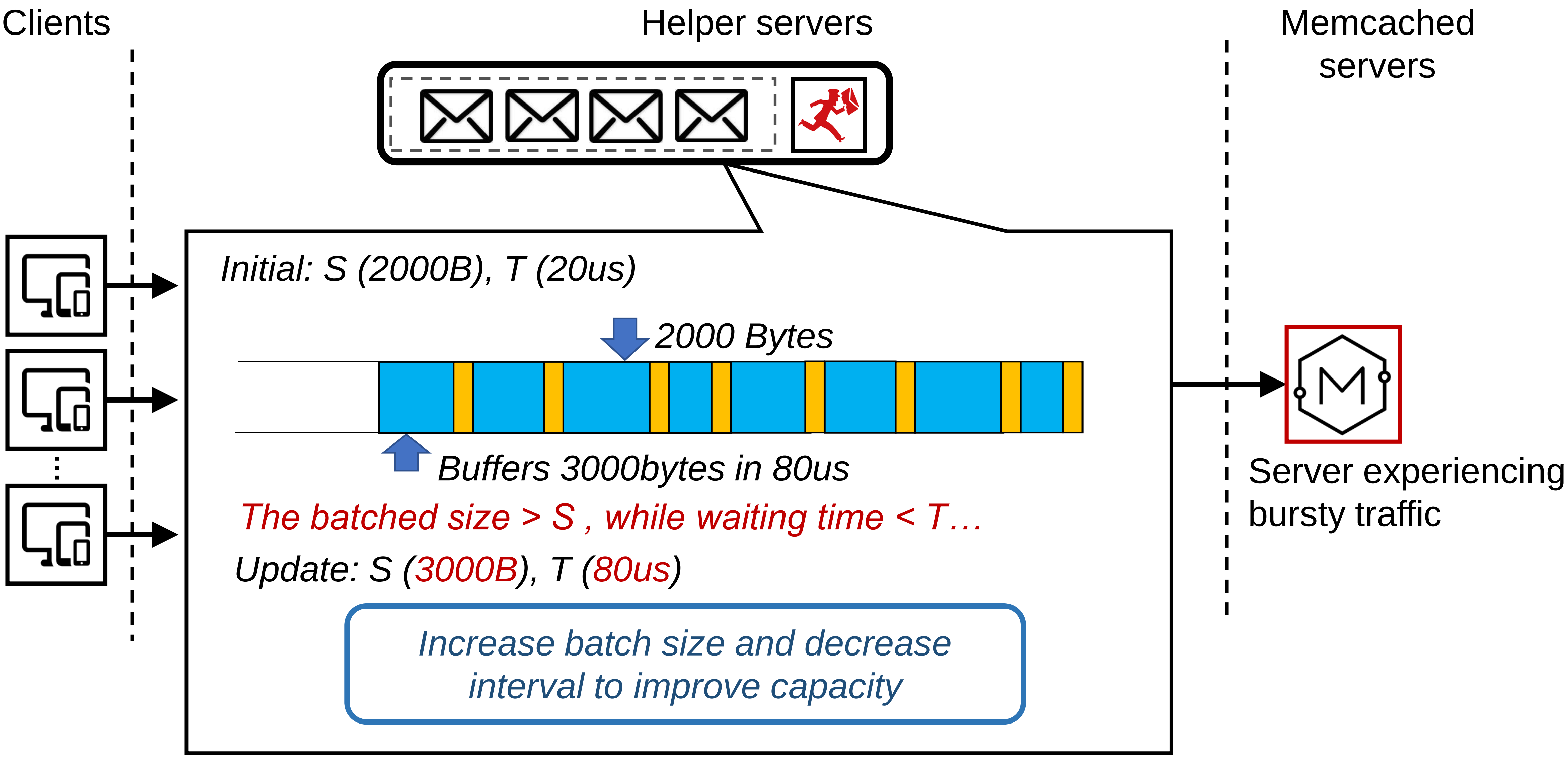
- *Batch size (S) and batch interval (T)*
- *Collect and buffer packets from clients in a loop*
- *Record buffer size and waiting time*
- *Update S and T based on buffer size and waiting time*

Lower bound of batch interval is 10us

Lower bound of batch size is MTU

Server experiencing bursty traffic

Is batching in helpers adaptive?



Clients

Helper servers

Memcached servers

Initial: S (2000B), T (20us)

2000 Bytes

Buffers 3000bytes in 80us

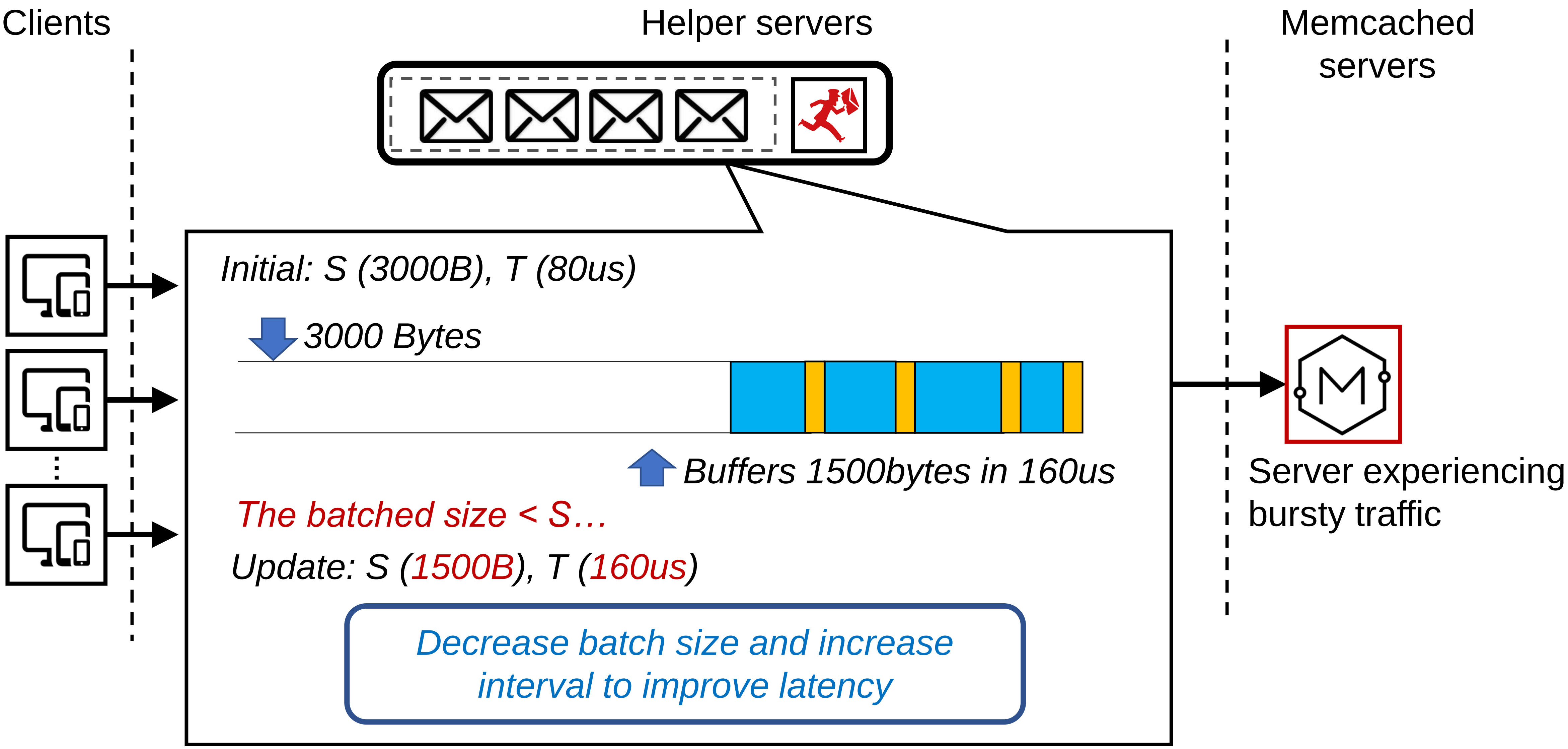
The batched size > S , while waiting time < T…

Update: S (3000B), T (80us)

Increase batch size and decrease interval to improve capacity

Server experiencing bursty traffic

Is batching in helpers adaptive?



Clients

Helper servers

Memcached servers

Initial: S (3000B), T (80us)

3000 Bytes

Buffers 1500bytes in 160us

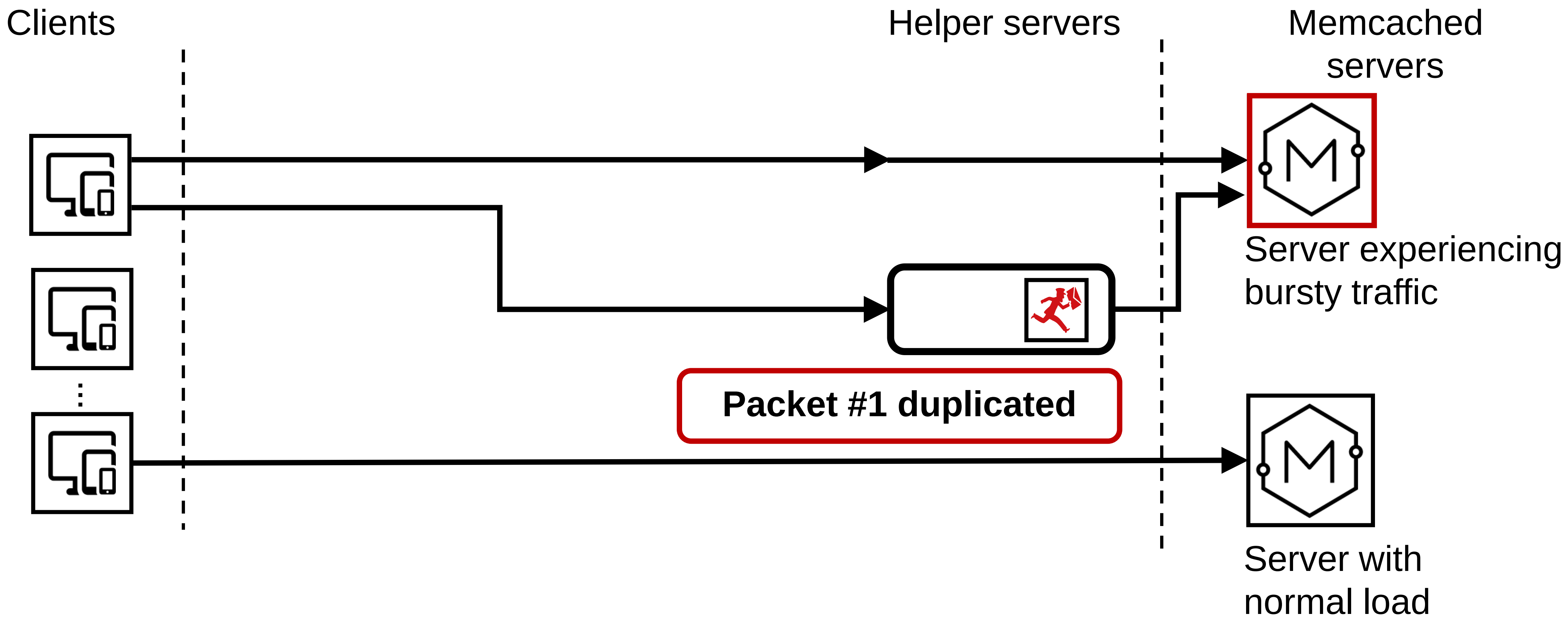
The batched size < S…

Update: S (1500B), T (160us)

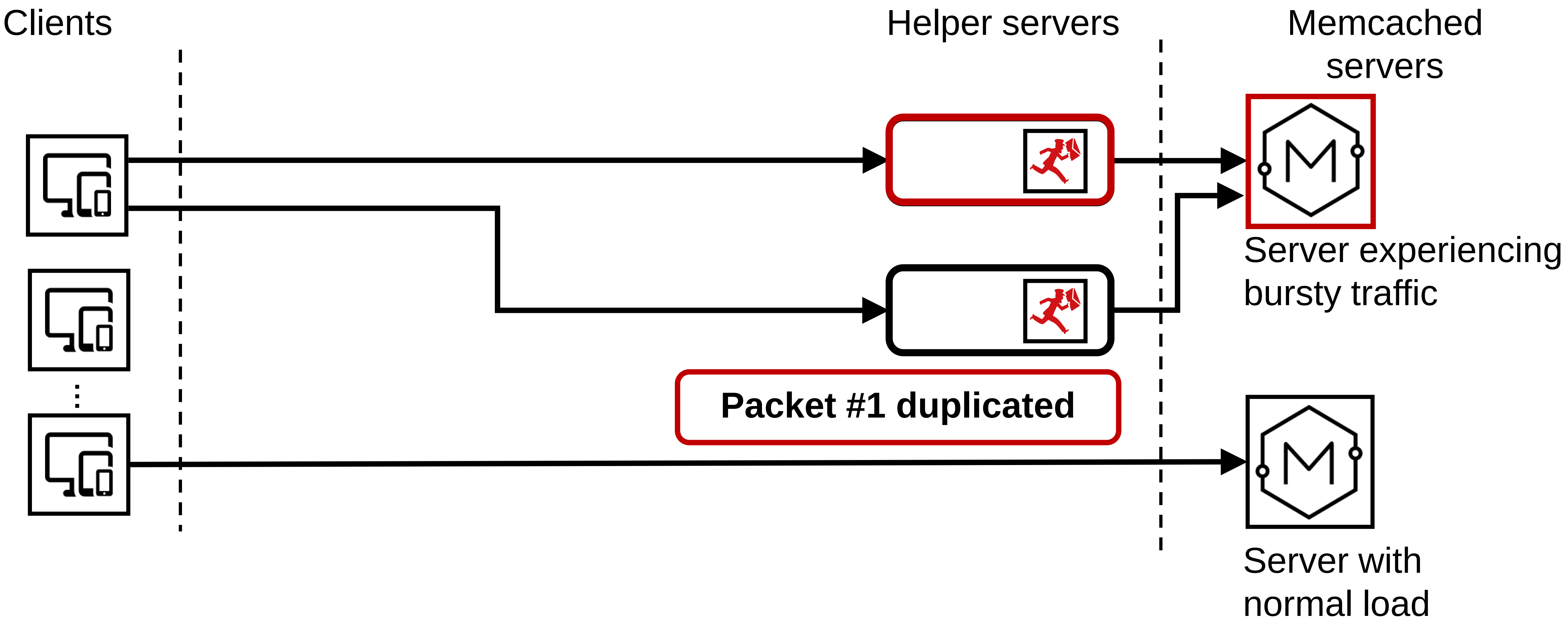
Decrease batch size and increase interval to improve latency

Server experiencing bursty traffic

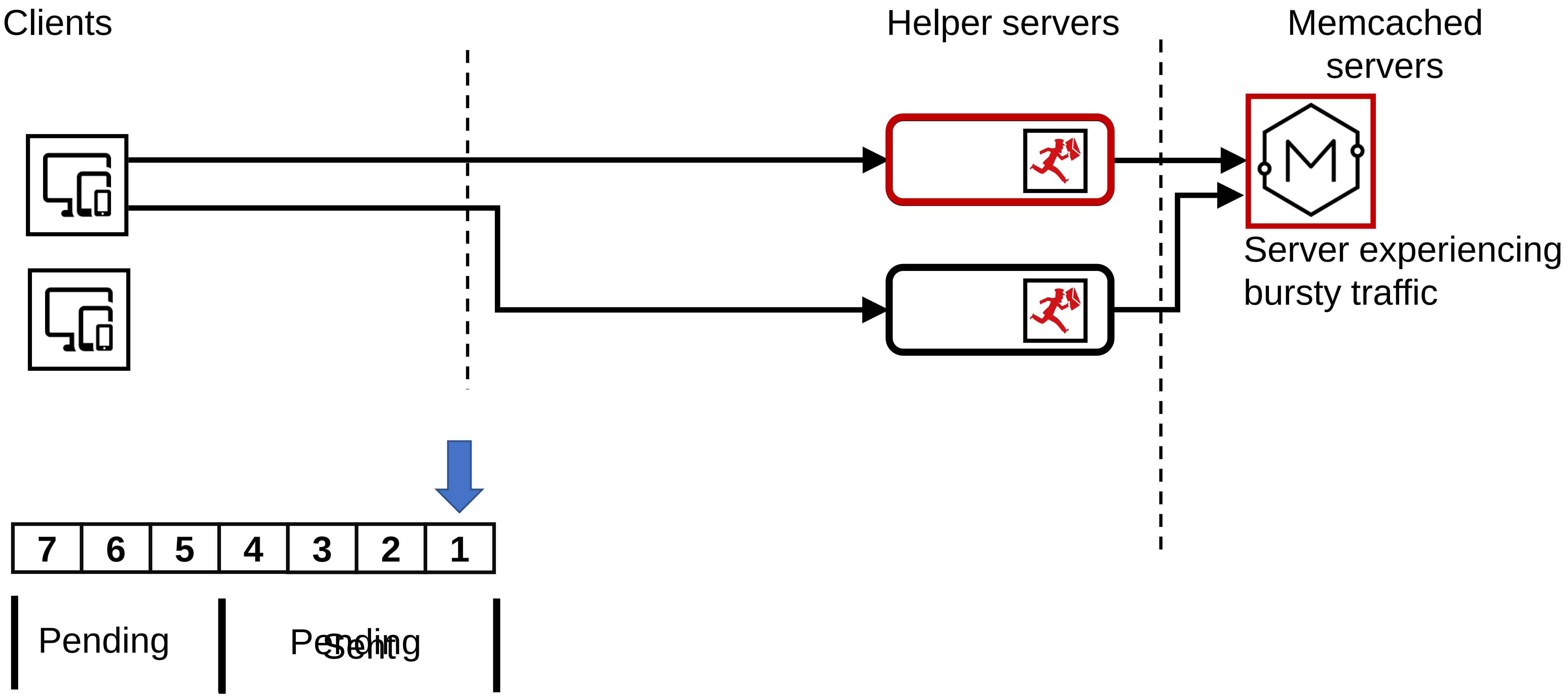
Connect to helpers



Switch to other helpers



Migrate connection after pending packets are received



7	6	5	4	3	2	1

Further optimization

□ Better load balancing strategy

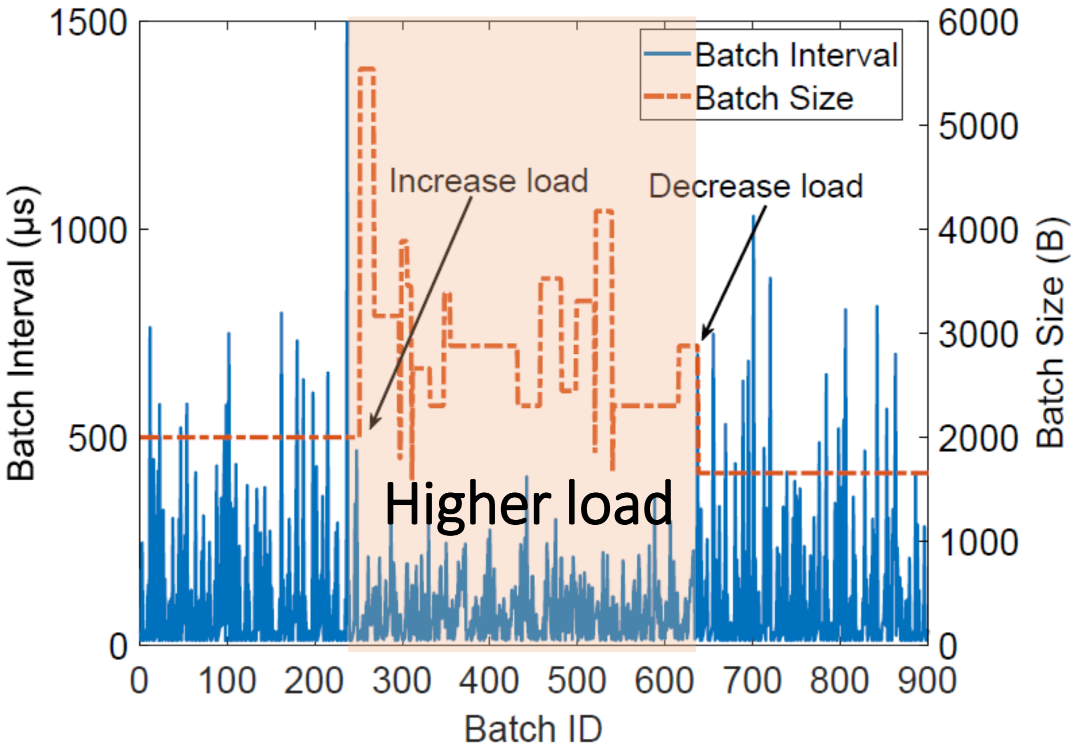
□ Information communication

- Better solution to update utilization of helpers in clients
- Better solution of informing clients to disconnect from helpers

□ Management of large scale helpers

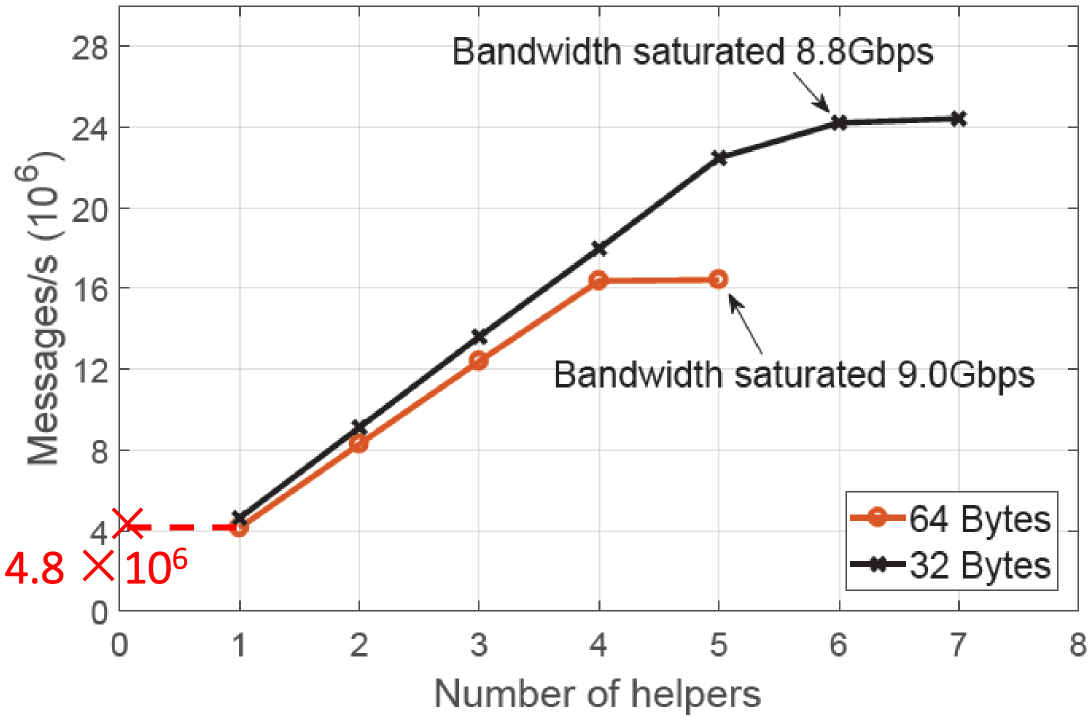
- How to startup and shutdown helpers

Adaptive batching



Higher load → Larger batch size & smaller batch interval
Lower load → Smaller batch size & Larger batch interval

Performance of PostMan



A single helper node can process about **9.6 million** small messages (2×4.8 million) per second