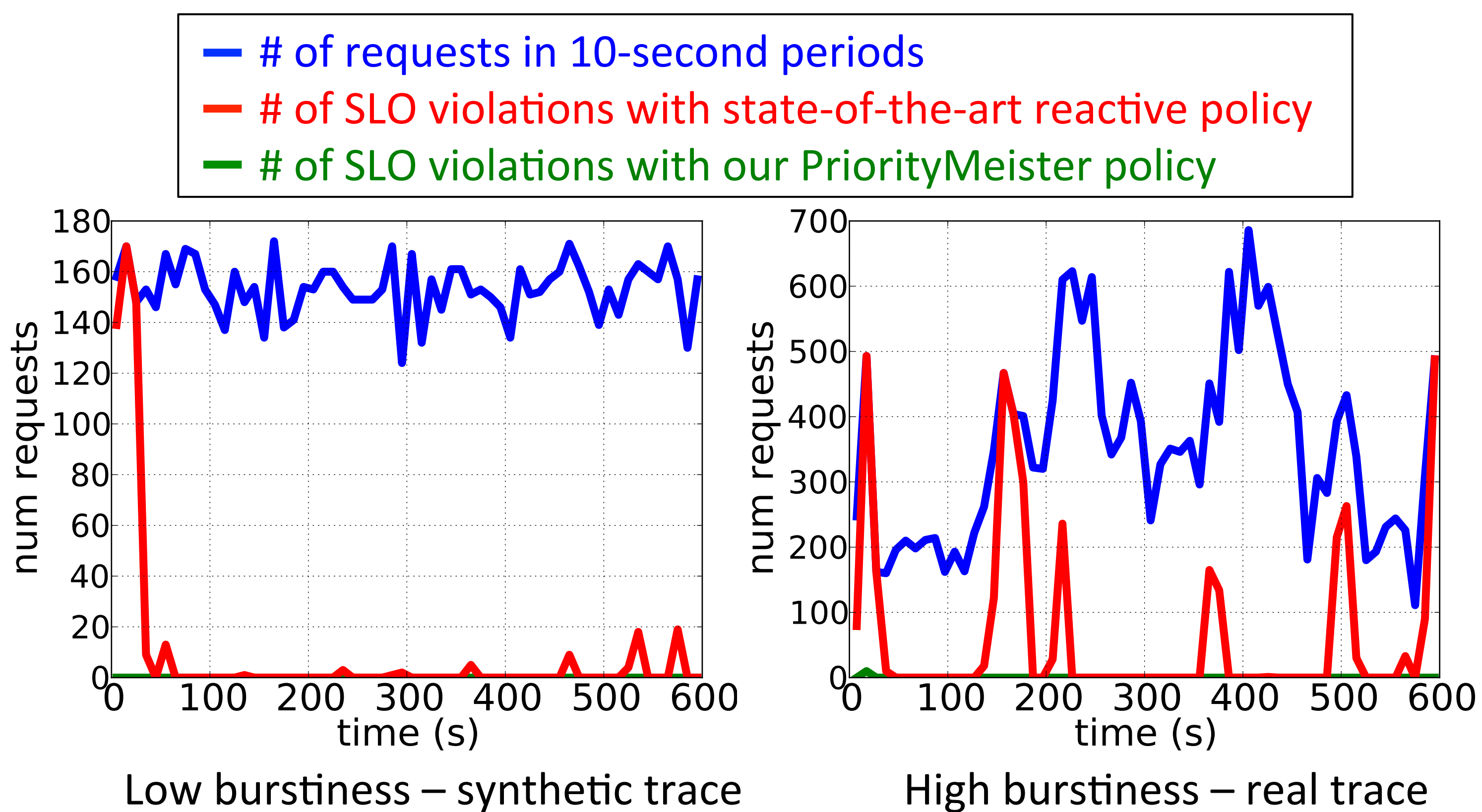


# PriorityMeister: Tail Latency QoS for Shared Networked Storage

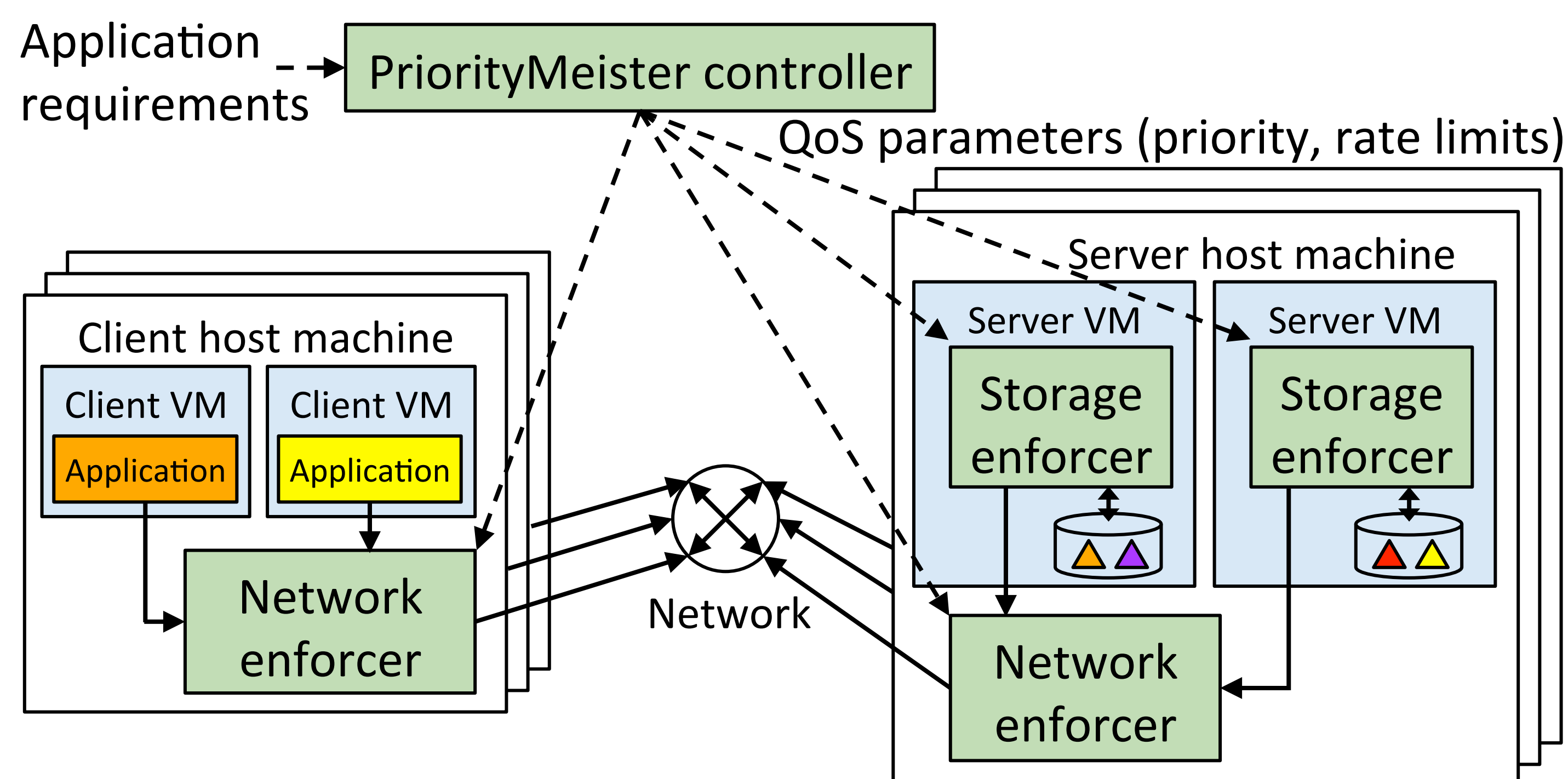
Timothy Zhu\* Alexey Tumanov\* Michael A. Kozuch† Mor Harchol-Balder\* Greg Ganger\* CMU\* Intel Labs†

## Problem/Motivation

- Goal: Meet per-application tail latency SLOs
  - in shared networked storage infrastructures
  - with bursty applications
- Challenges:
  - End-to-end latency is affected by storage & network
  - Bursts affect tail latencies of workloads sharing system



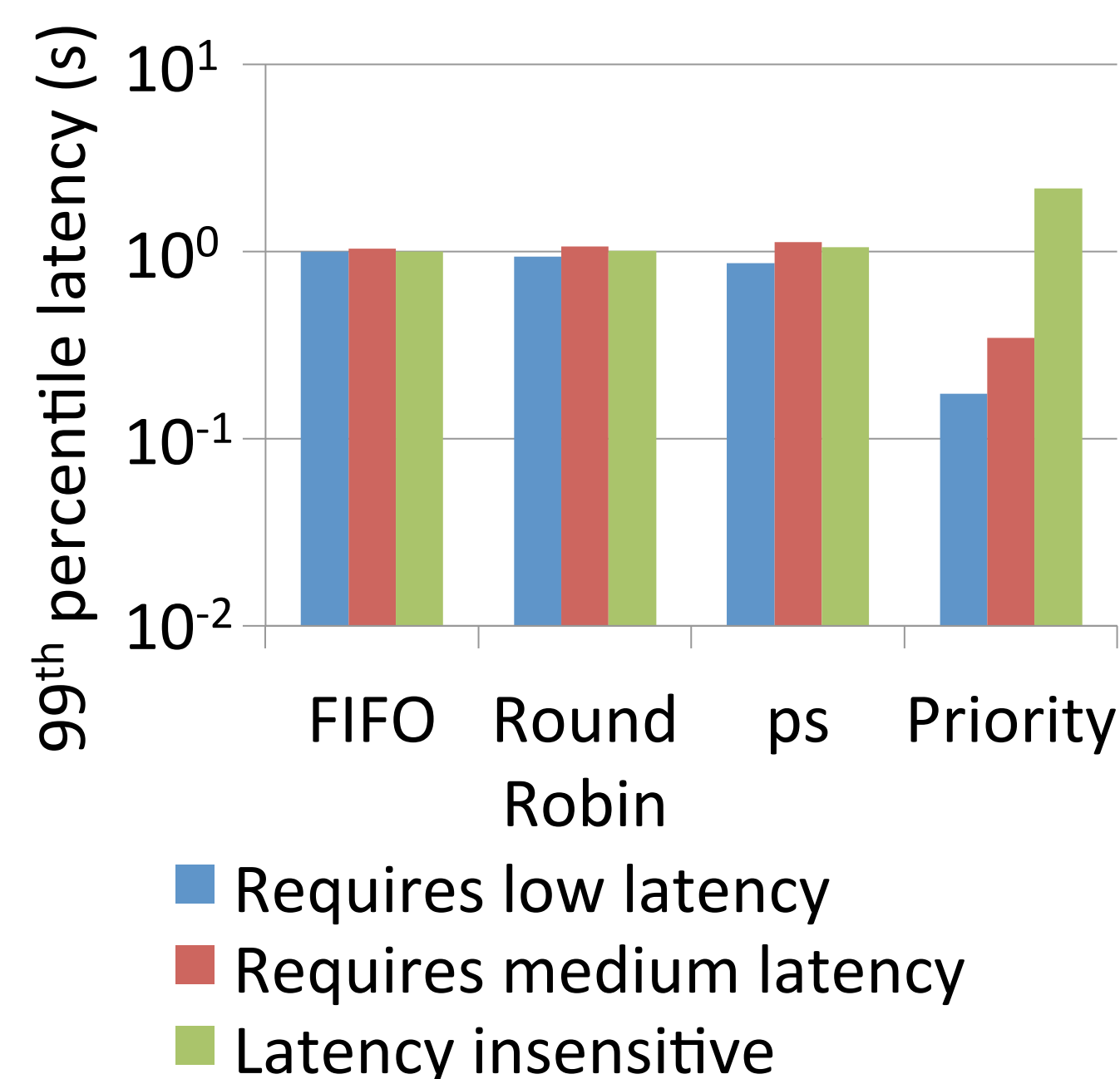
## System architecture



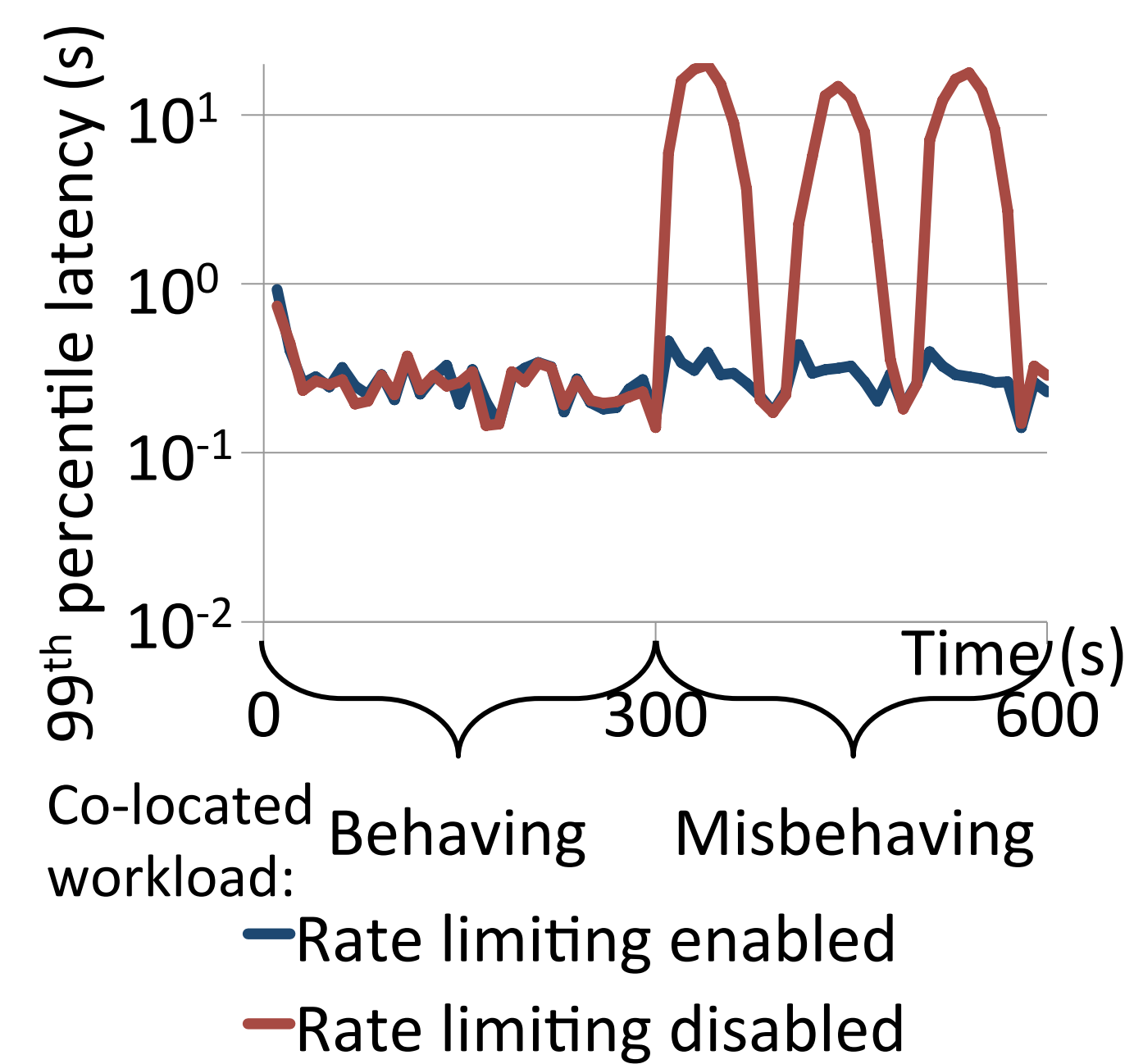
## QoS enforcement

- Priority – reduces latency for workloads that care most
- Rate limits – protects workloads from misbehaving workloads

### Priority – latency differentiation

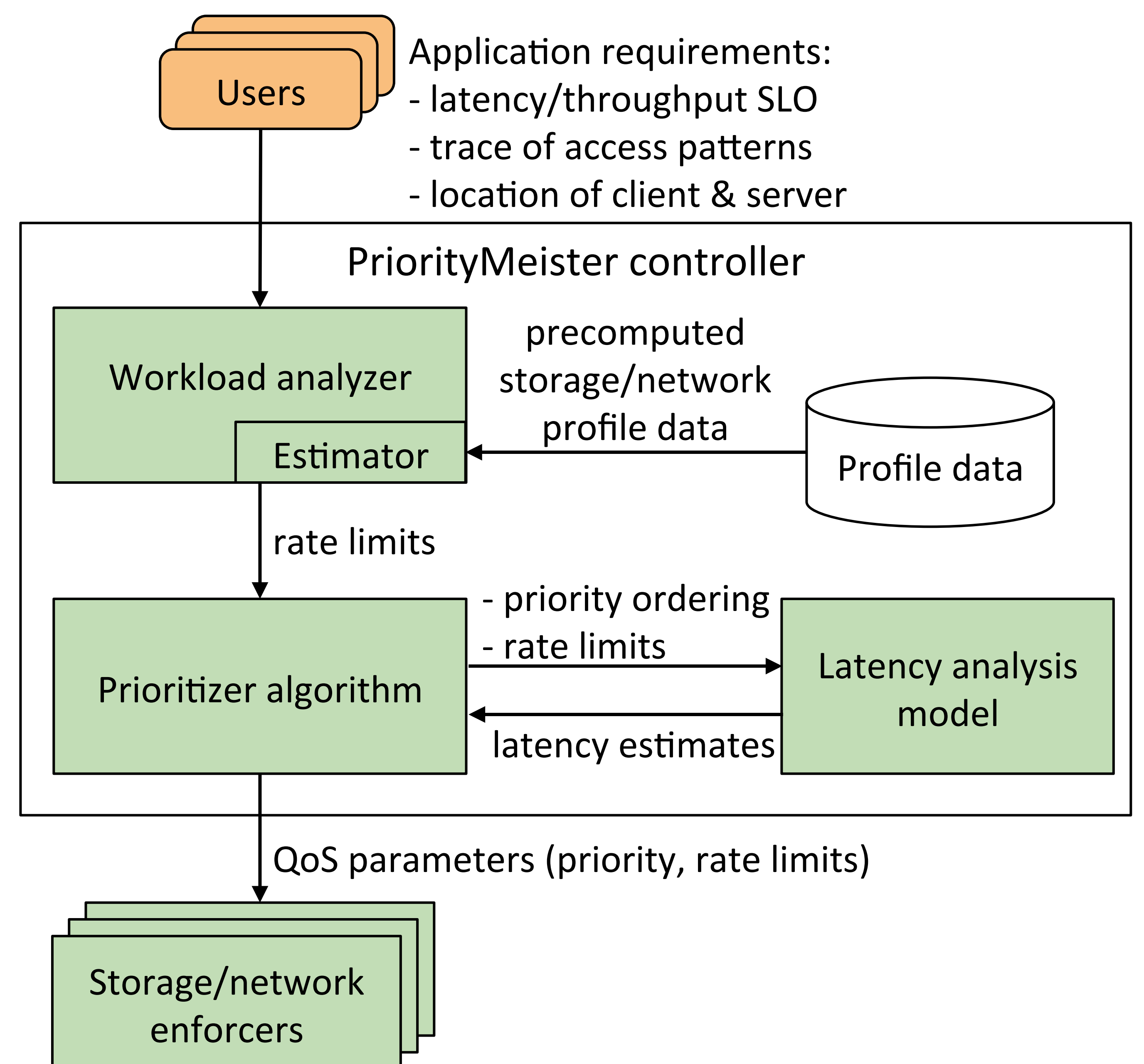


### Rate limiting - performance isolation



## PriorityMeister controller design

- Analyzes workloads to identify bottlenecks in storage & network
- Configures priorities and rate limits to meet SLOs



## Results

- PriorityMeister (PM) outperforms Cake, bySLO, EDF, ps
- Only PM meets SLOs for all workloads at tail

