High-resolution Measurement of Data Center Microbursts

Qiao Zhang (University of Washington)
Vincent Liu (University of Pennsylvania)
Hongyi Zeng (Facebook)
Arvind Krishnamurthy (University of Washington)

Networks are Fast, Measurements are not...

Data center networks are getting faster

- 100Gbps, ~100 ns to process a packet, 10-100 µs RTT

But measurement frameworks are not keeping up

- **SNMP counters** (e.g. bytes sent or drops) typically collected every couple minutes

- **Packet sampling** (sFlow or iptables) typically at low sampling rate, e.g. 1/30k

Networks are Fast, Measurements are not...

Data center networks are getting faster

- 100Gbps, ~100 ns to process a packet, 10-100 µs RTT

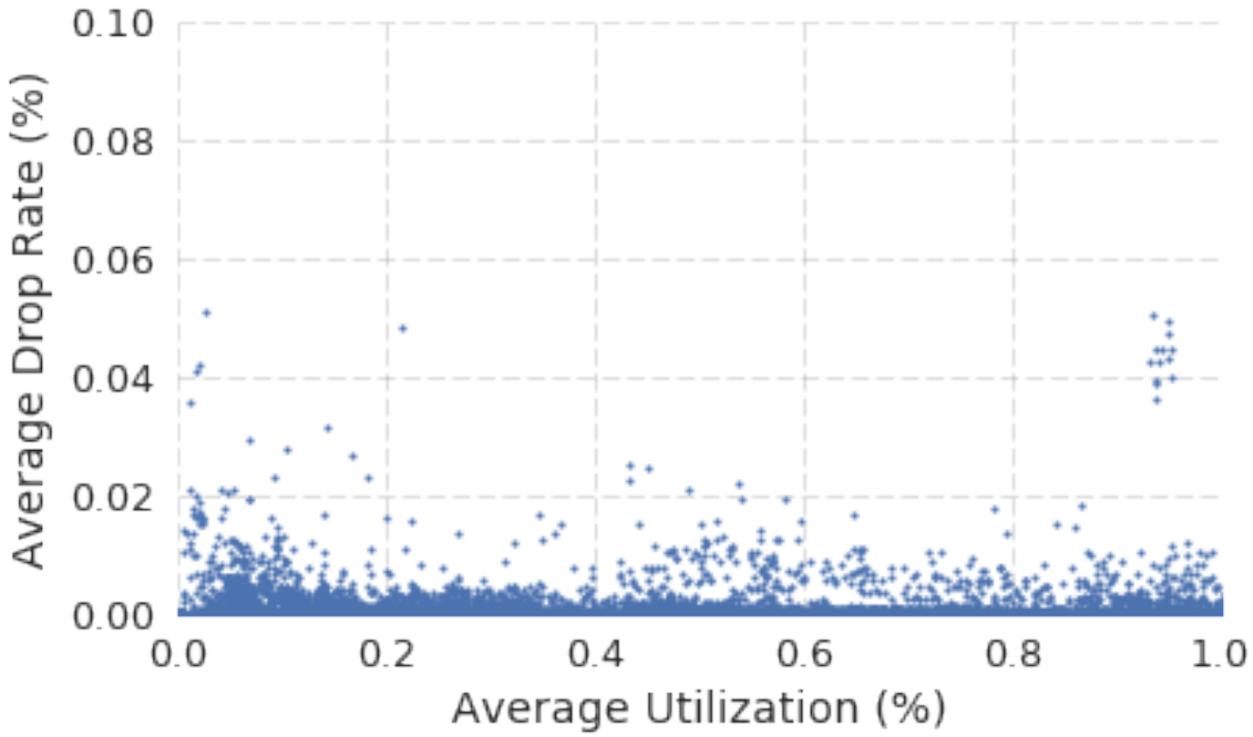
But measurement frameworks are not keeping up

- **SNMP counters** (e.g. bytes sent or drops) typically collected every couple minutes

- **Packet sampling** (sFlow or iptables) typically at low sampling rate, e.g. 1/30k

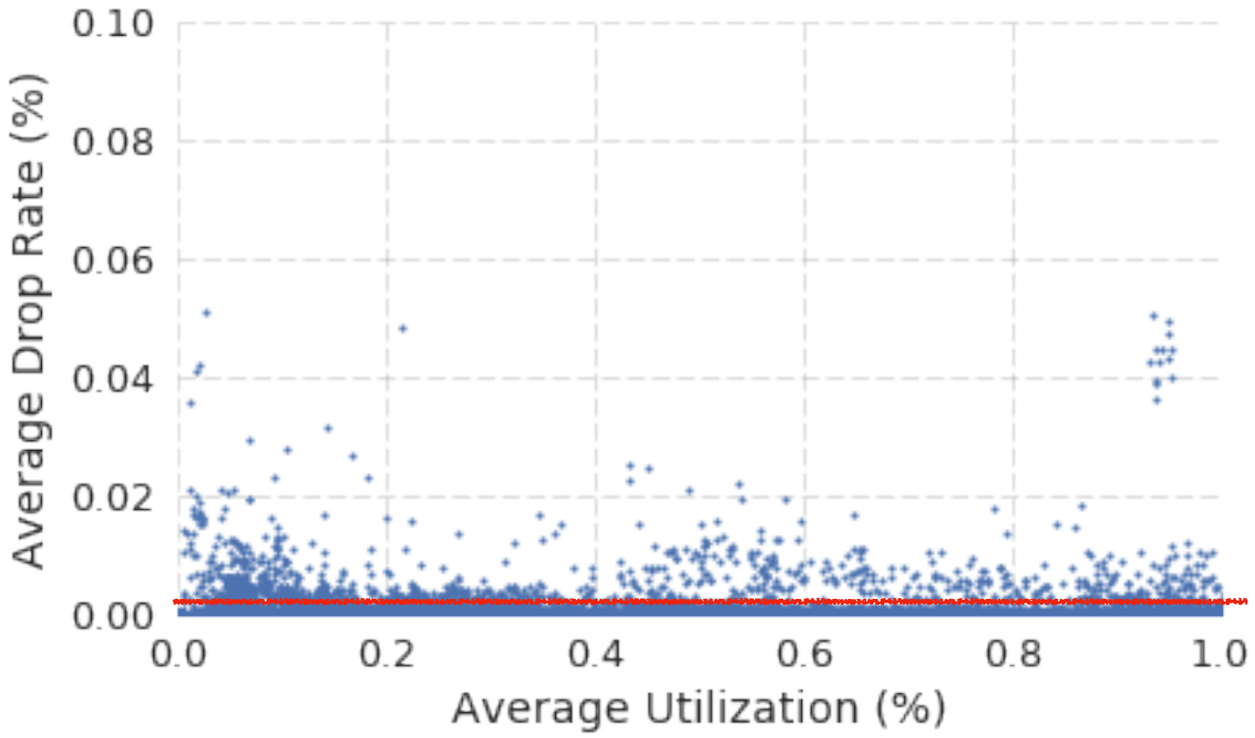
Too coarse-grained !

The Case for High Resolution



- Packet drop correlates poorly with utilization at 4 minute granularity
- 4 minute granularity hides short-term traffic spikes
- Need high-resolution to reveal finer-grained behaviors

The Case for High Resolution

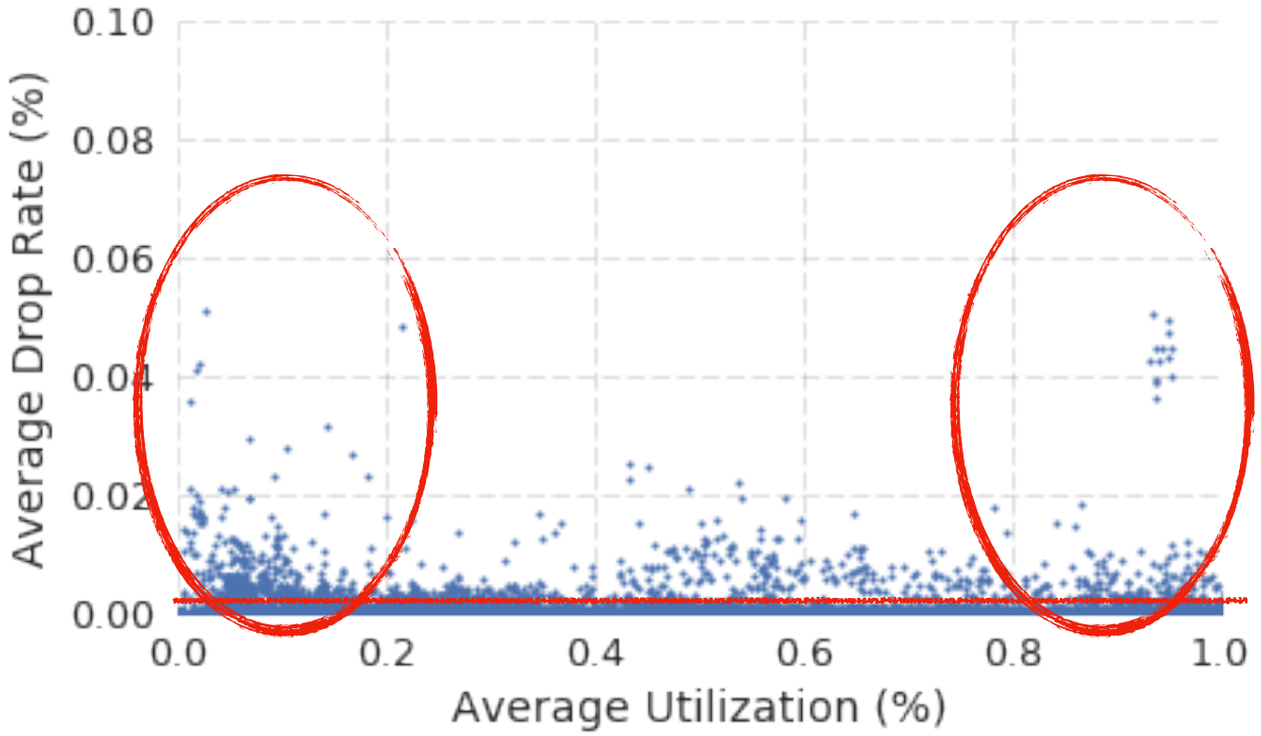


drop rate generally very low

- Packet drop correlates poorly with utilization at 4 minute granularity
- 4 minute granularity hides short-term traffic spikes
- Need high-resolution to reveal finer-grained behaviors

The Case for High Resolution

unusual drop rates at both low and high utilization



drop rate generally very low

- Packet drop correlates poorly with utilization at 4 minute granularity
- 4 minute granularity hides short-term traffic spikes
- Need high-resolution to reveal finer-grained behaviors

Roadmap

Mechanism

- It is possible to do high resolution measurements on today's switches

Results

- Many if not most traffic bursts are very short-lived

High-resolution Counter Collection Framework

We designed a **high-resolution** counter collection framework

- Switch CPUs poll ASIC registers with microsecond level latency
- Sample fast (~25 μs) while keeping sampling loss below 1%

We focus on three kinds of counters

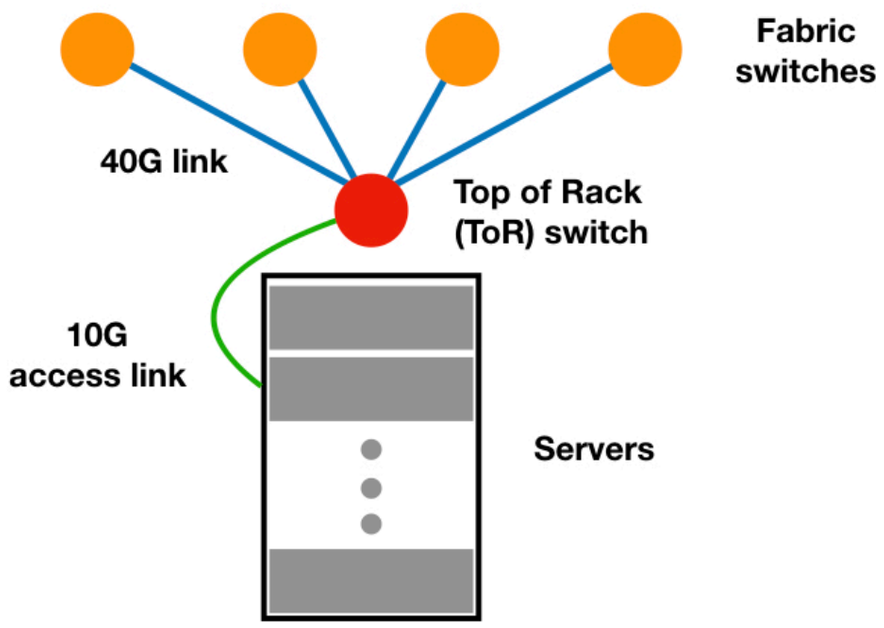
1. **Byte count**: cumulative and used to compute utilization
2. **Packet size**: a histogram of packet sizes
3. **Peak buffer occupancy**: for single port and shared pool

Deployment

- One of the largest data centers at Facebook with a 3-tier Clos network

- Only collect from ToRs due to deployment constraints

- 10Gbps server links and 4x40Gbps ToR uplinks



Workload and Methodology

- Mostly single-role racks
  - Web: handle user request, lookup with cache
  - Cache: handle k-v lookups, respond to Web servers
  - Hadoop: handle batched processing
- 30 racks in total: 10 racks for each app, over 24 hours
  - Sample a random 2-minute interval per hour, for 1TB+

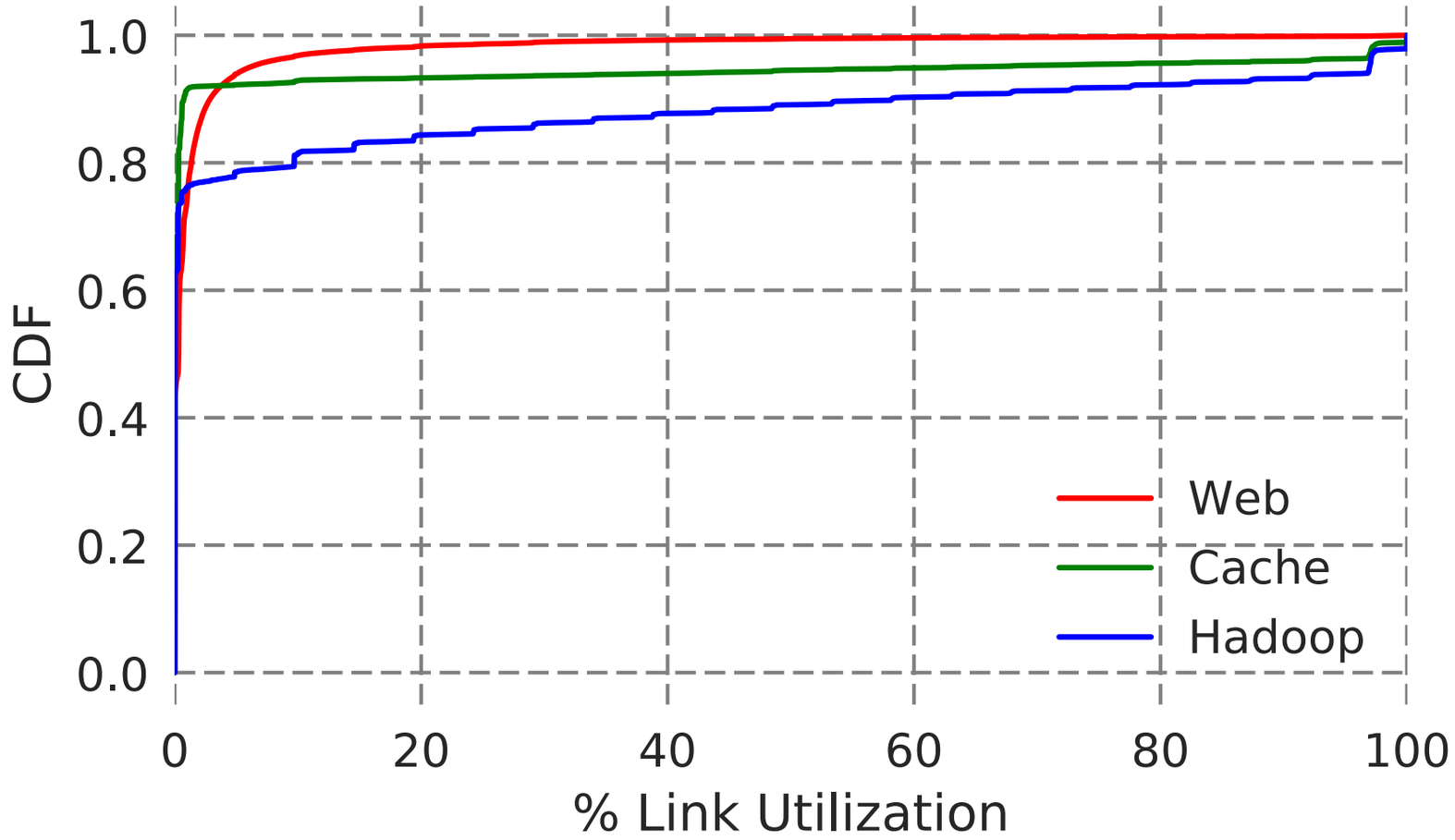
Microburst Measurements

Microburst:

a period of short-term high utilization (e.g. >50%)

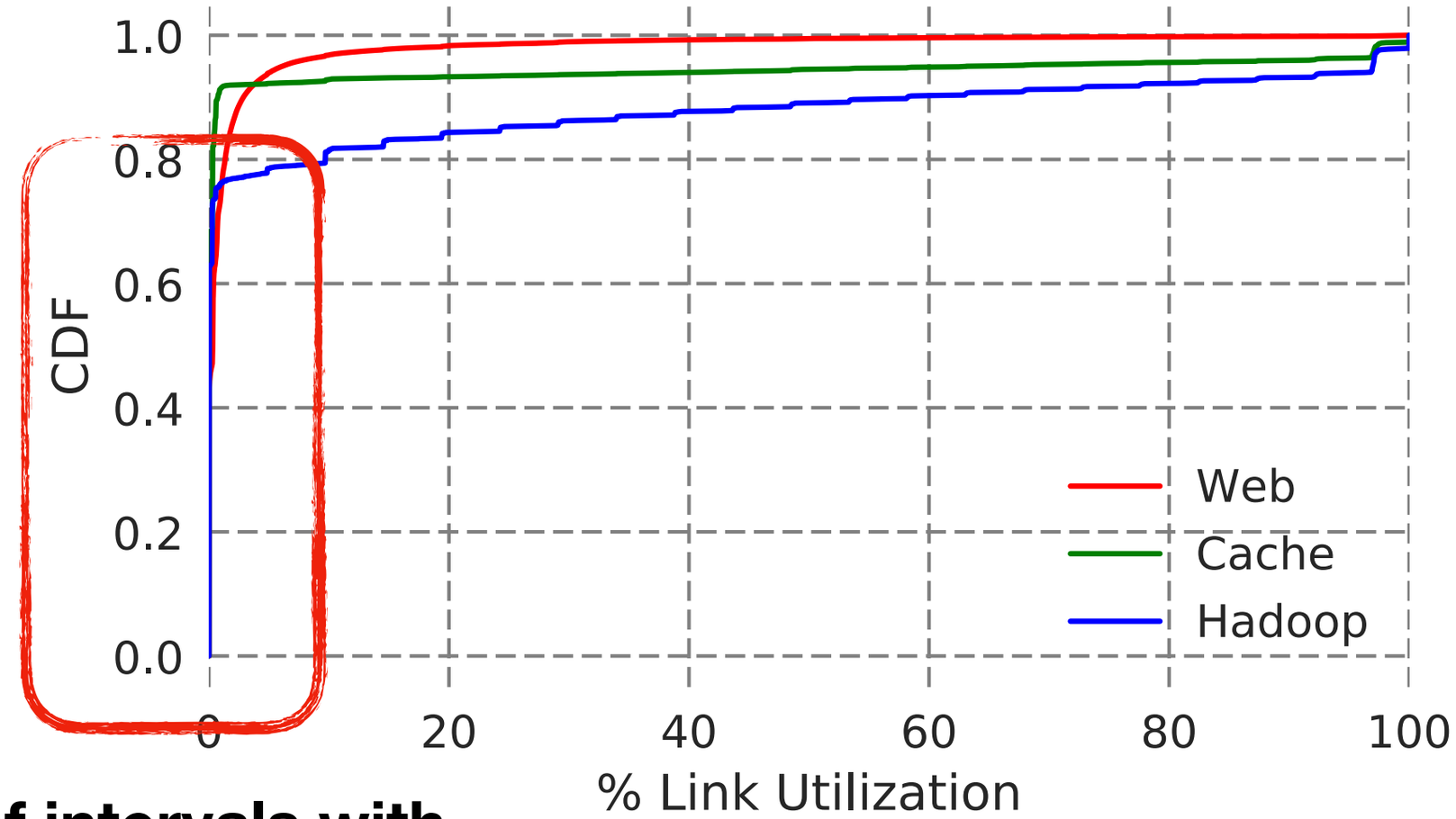
- How long do they last and how often do they occur?
- How much of congestion is caused by microbursts?
- Does network behavior differ significantly inside a burst?
- Are there synchronized behaviors during bursts?

Distribution of Link Utilization



25 µs

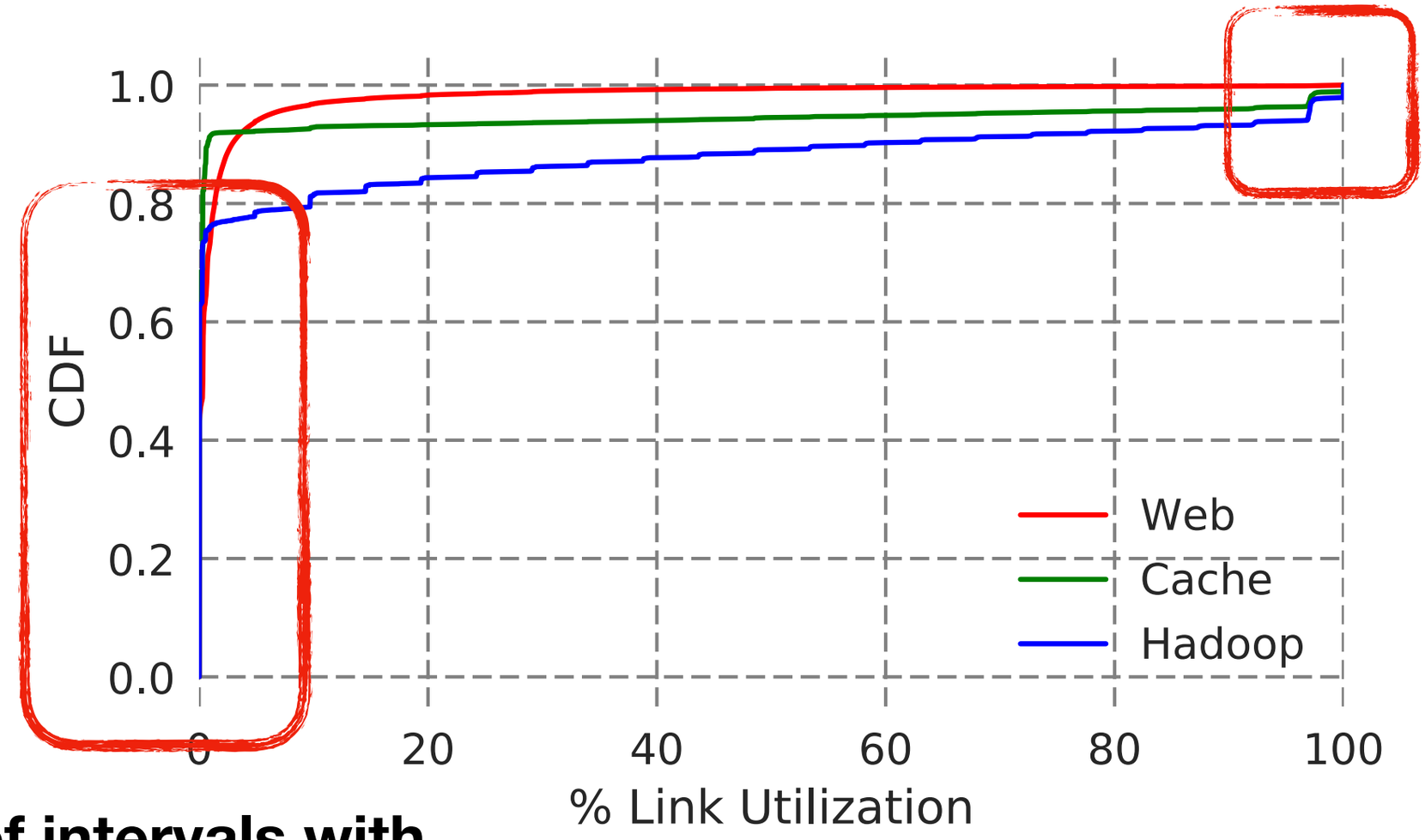
Distribution of Link Utilization



a lot of intervals with almost nothing happening

25 µs

Distribution of Link Utilization

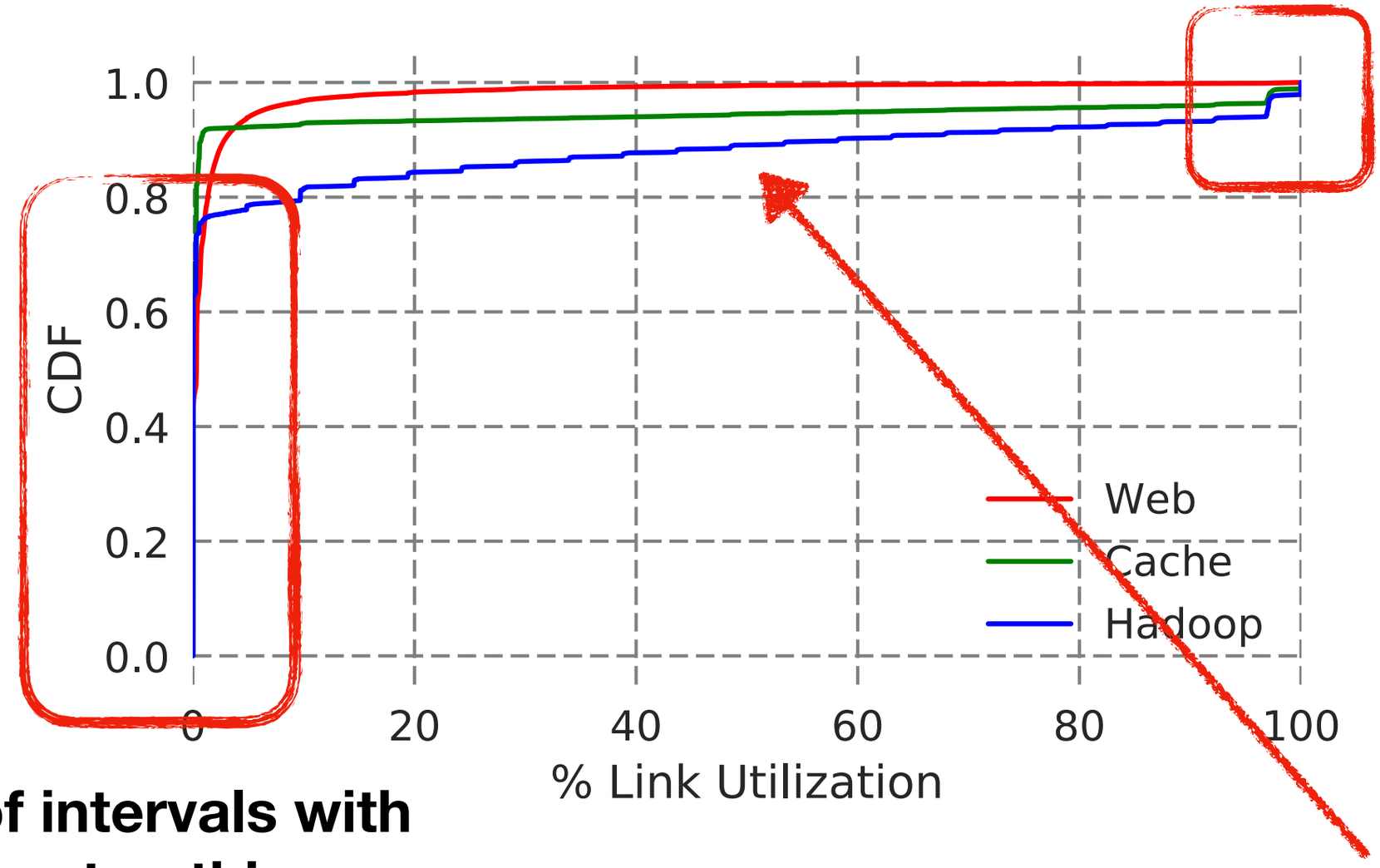


a few intervals have ~100% utilization

a lot of intervals with almost nothing happening

25 µs

Distribution of Link Utilization



a few intervals have ~100% utilization

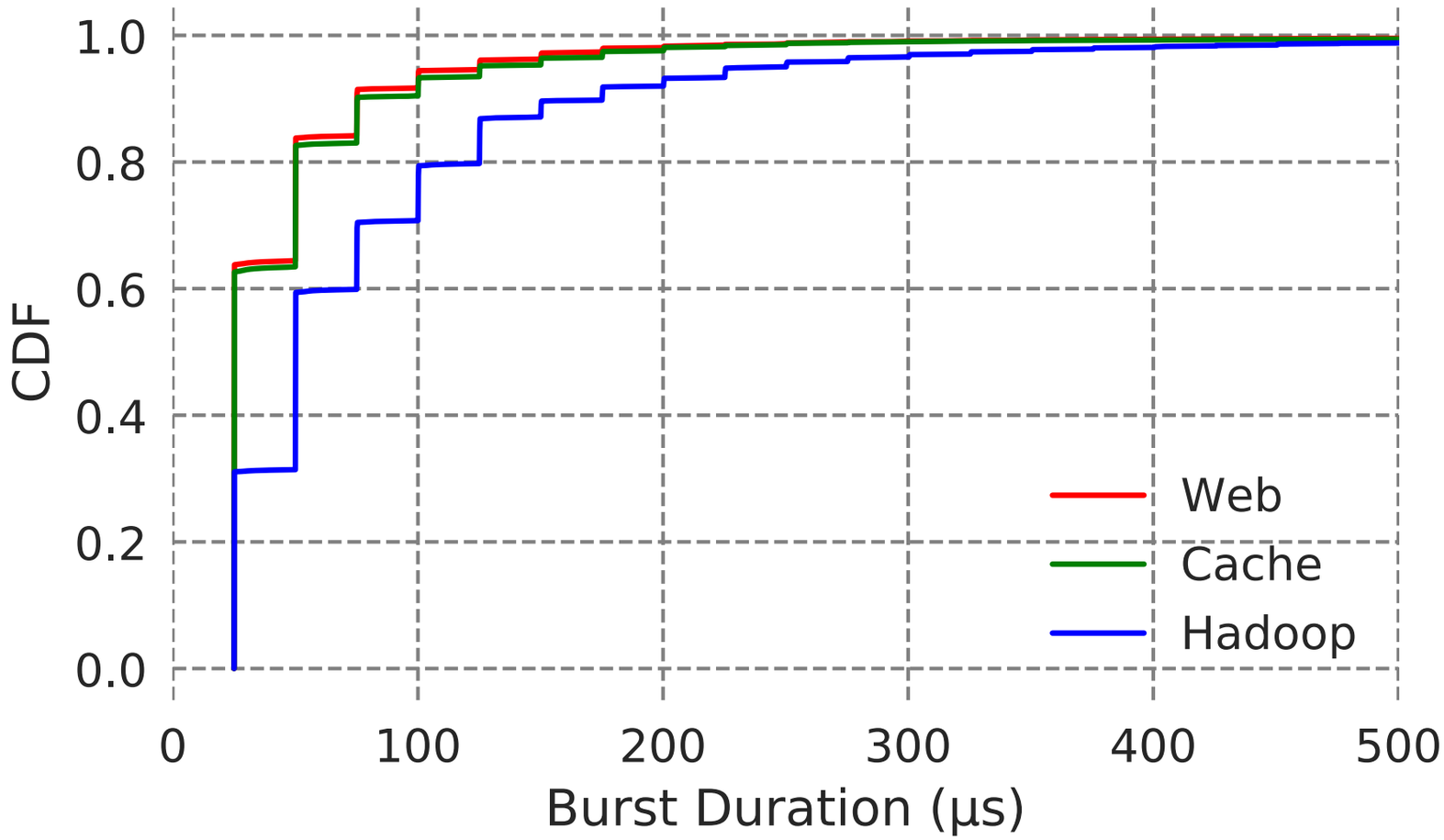
a lot of intervals with almost nothing happening

insensitive to 50% threshold

25 µs

Bursts are Short

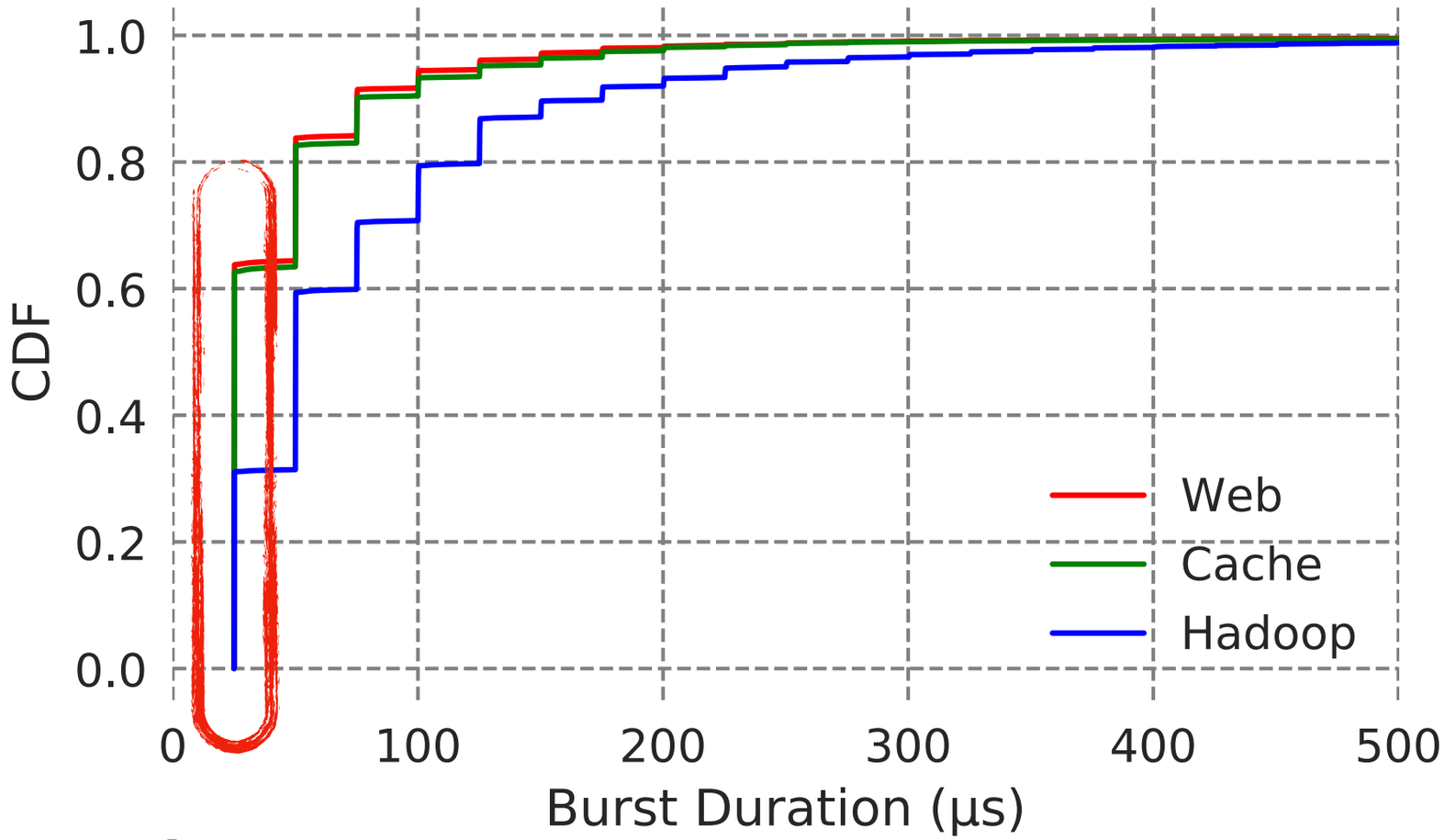
- **Burst**: an unbroken sequence of hot samples (> 50% util)



25 µs

Bursts are Short

- **Burst**: an unbroken sequence of hot samples (> 50% util)

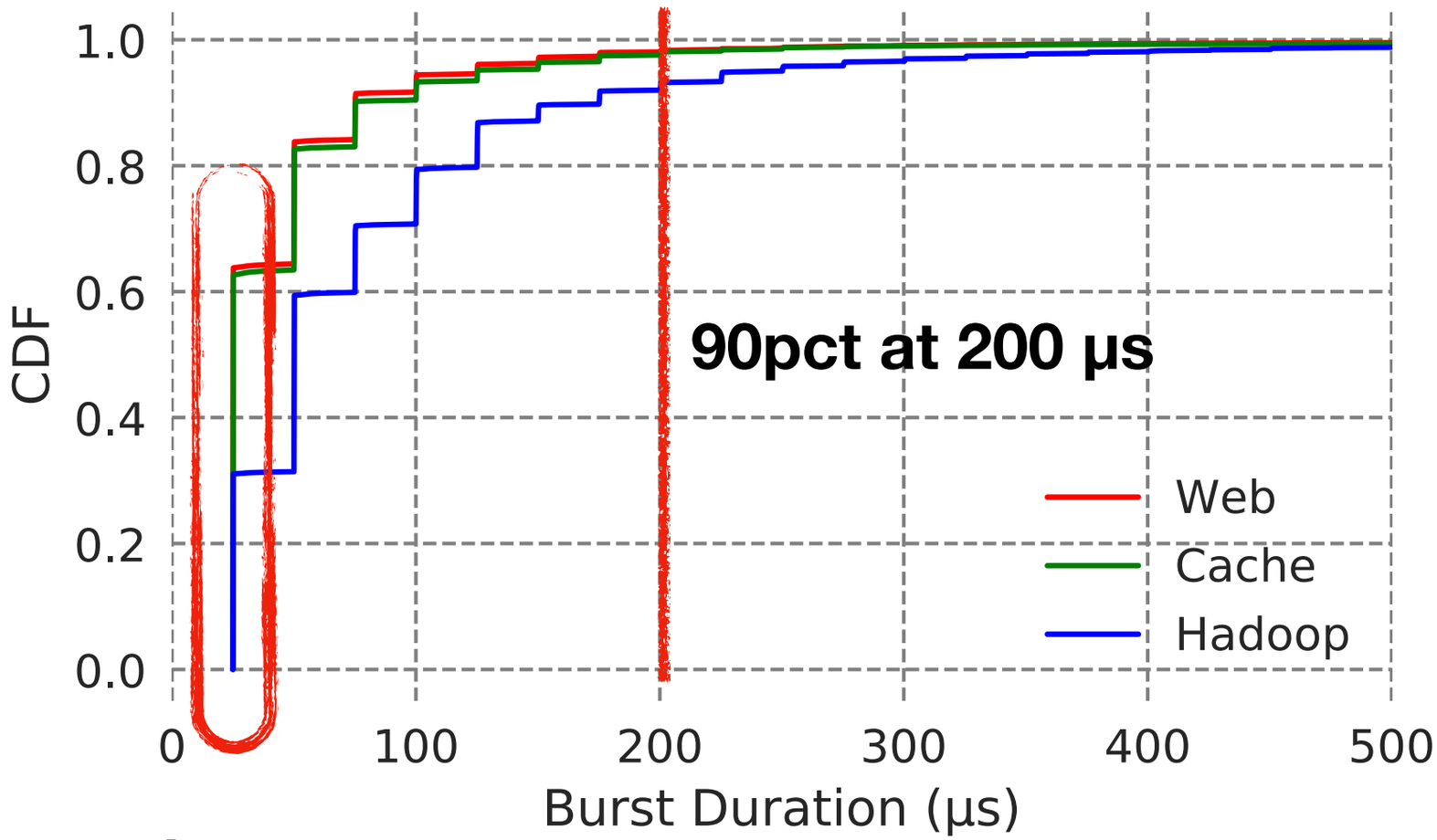


many bursts
last at most 25 µs

25 µs

Bursts are Short

- **Burst**: an unbroken sequence of hot samples (> 50% util)

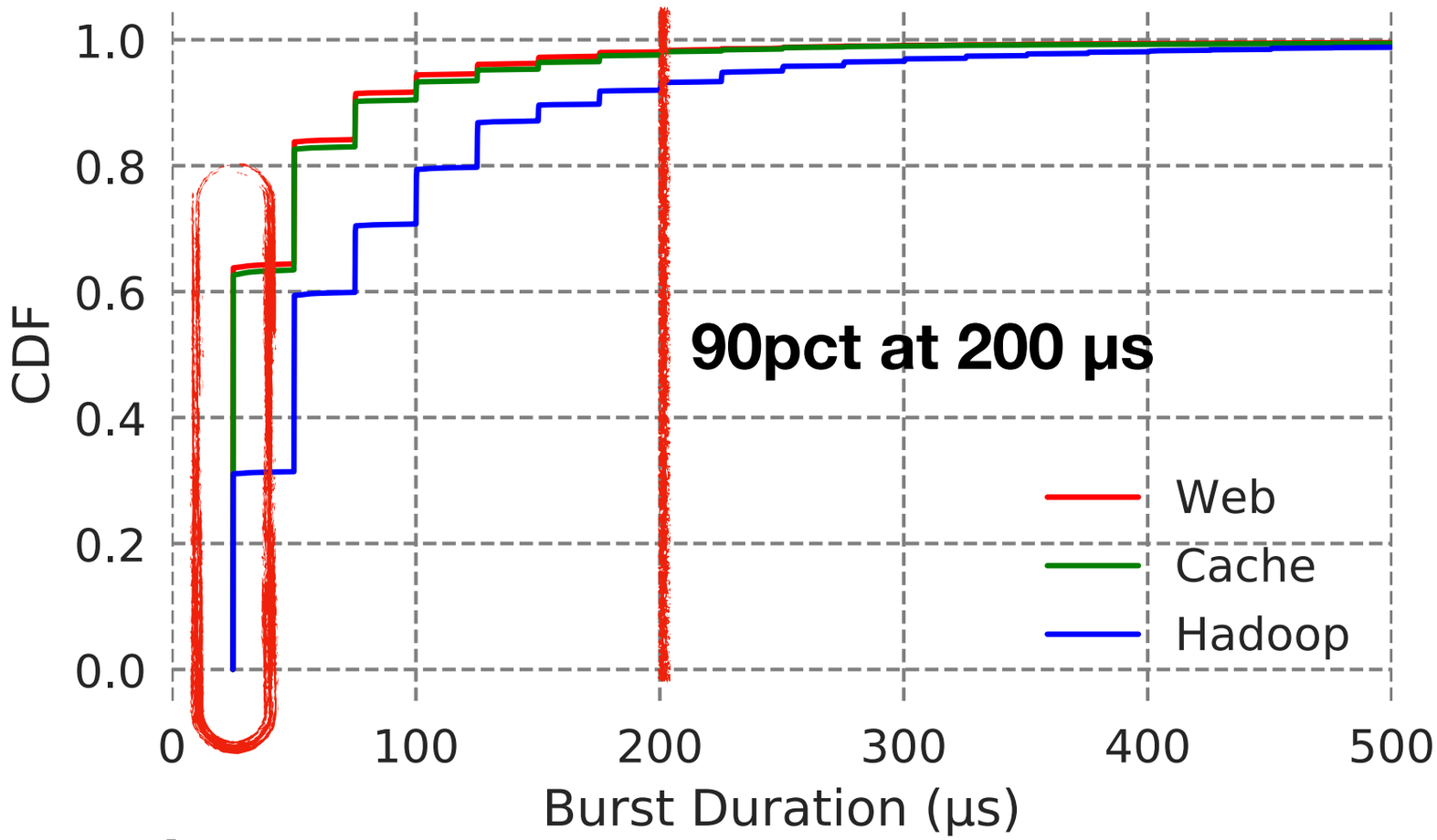


many bursts
last at most 25 µs

25 µs

Bursts are Short

- **Burst**: an unbroken sequence of hot samples (> 50% util)

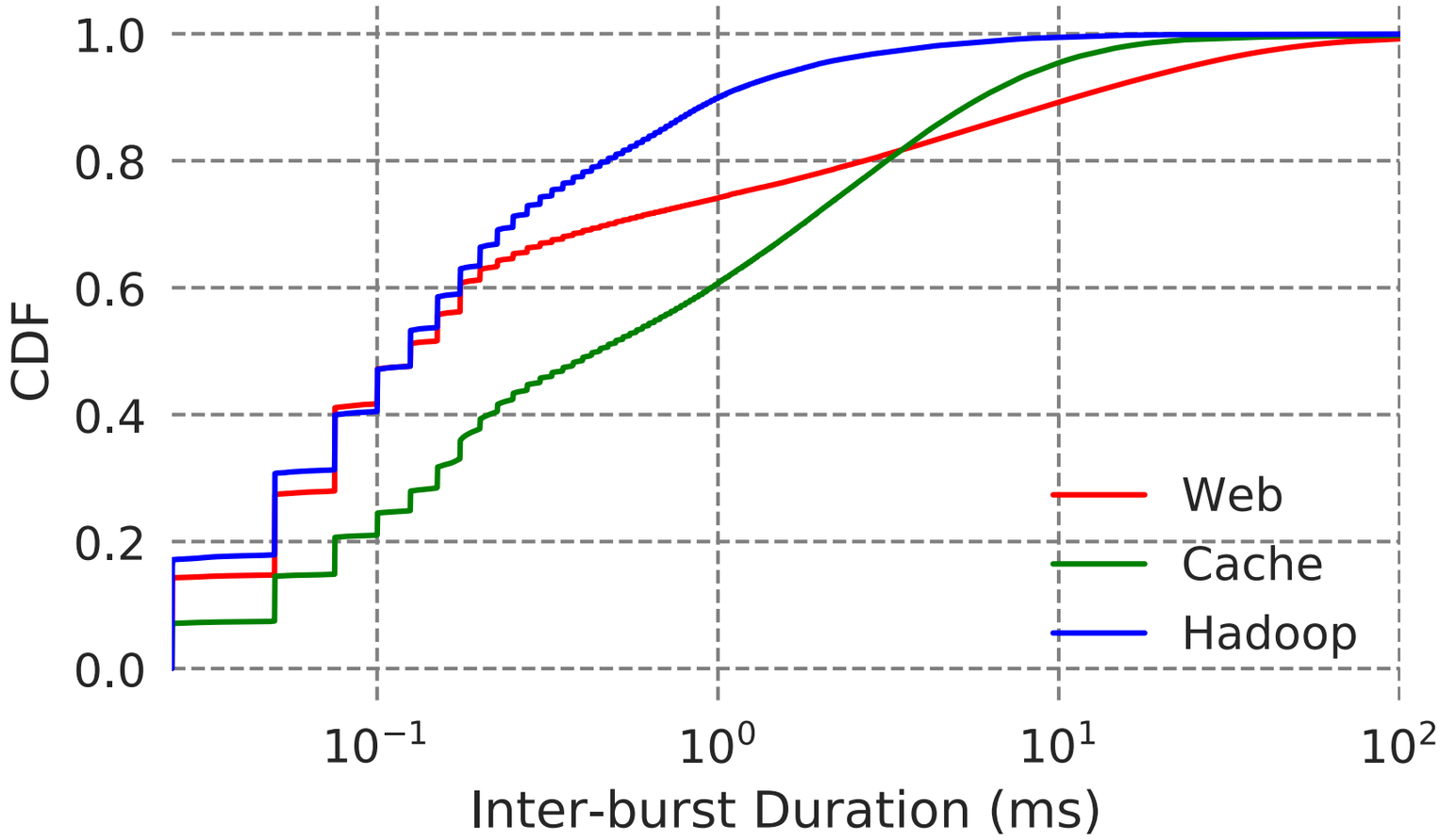


many bursts
last at most 25 µs

Almost all congestion is short-lived

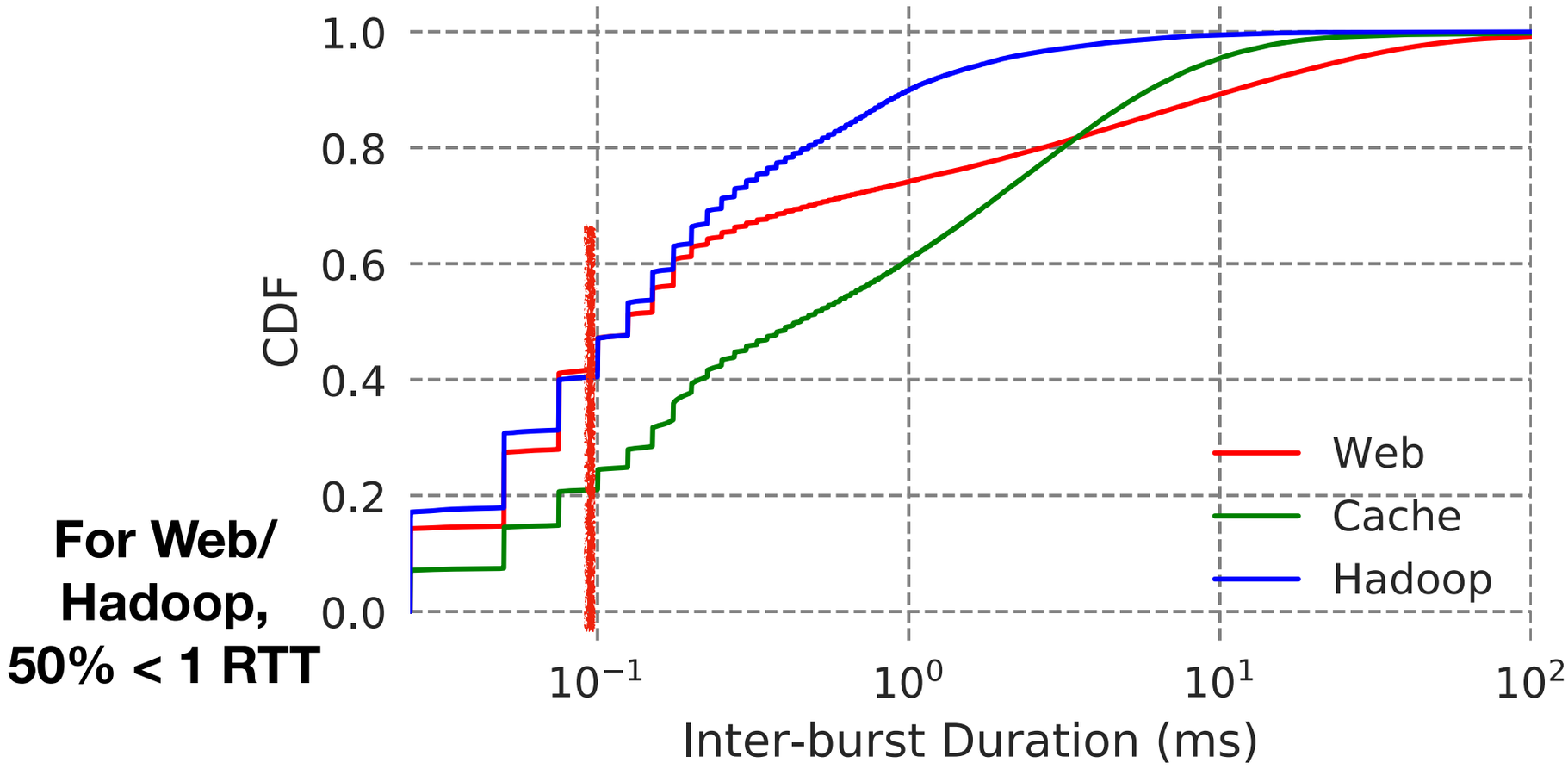
25 µs

Time between Bursts



25 µs

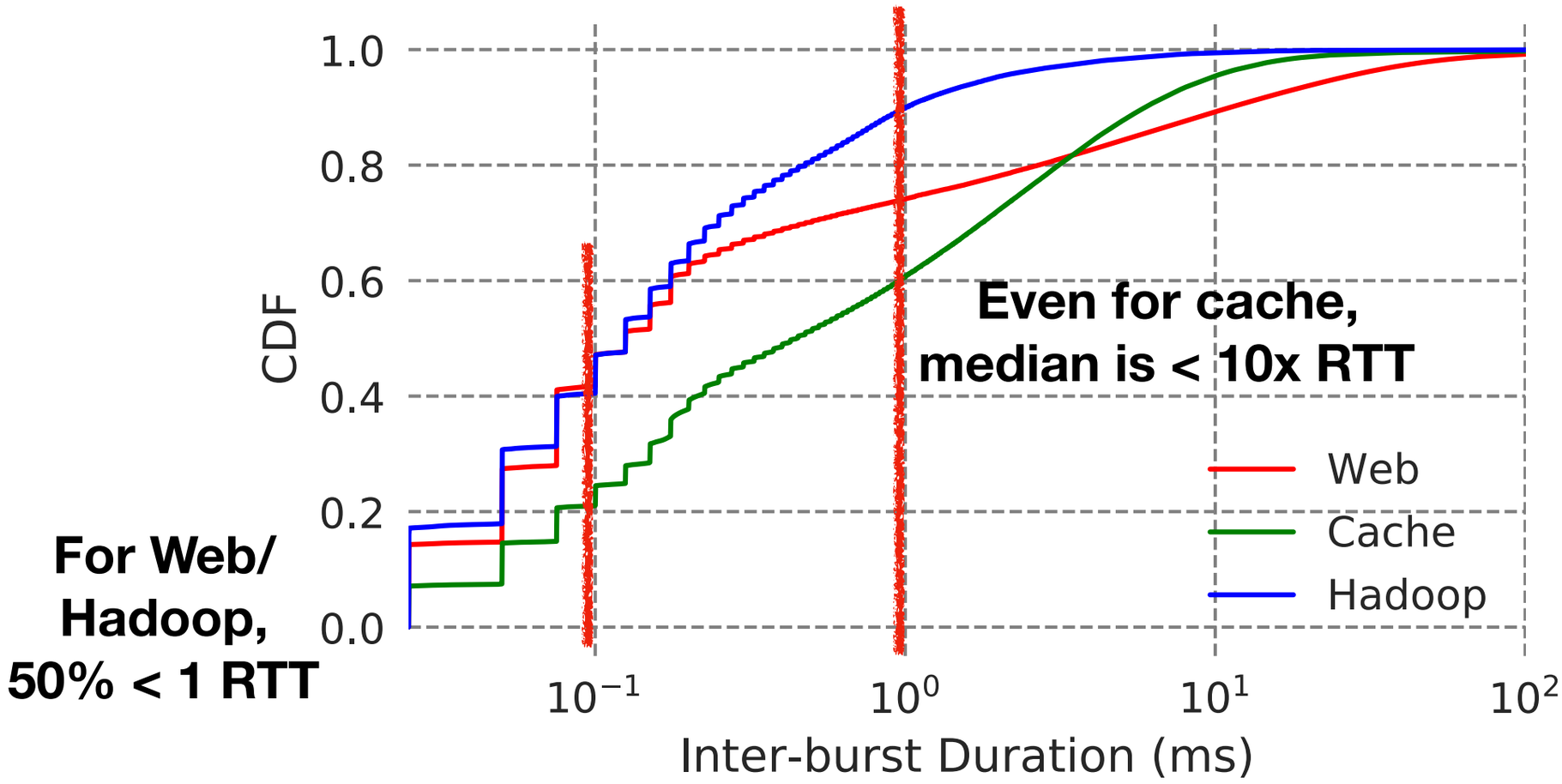
Time between Bursts



For Web/ Hadoop, 50% < 1 RTT

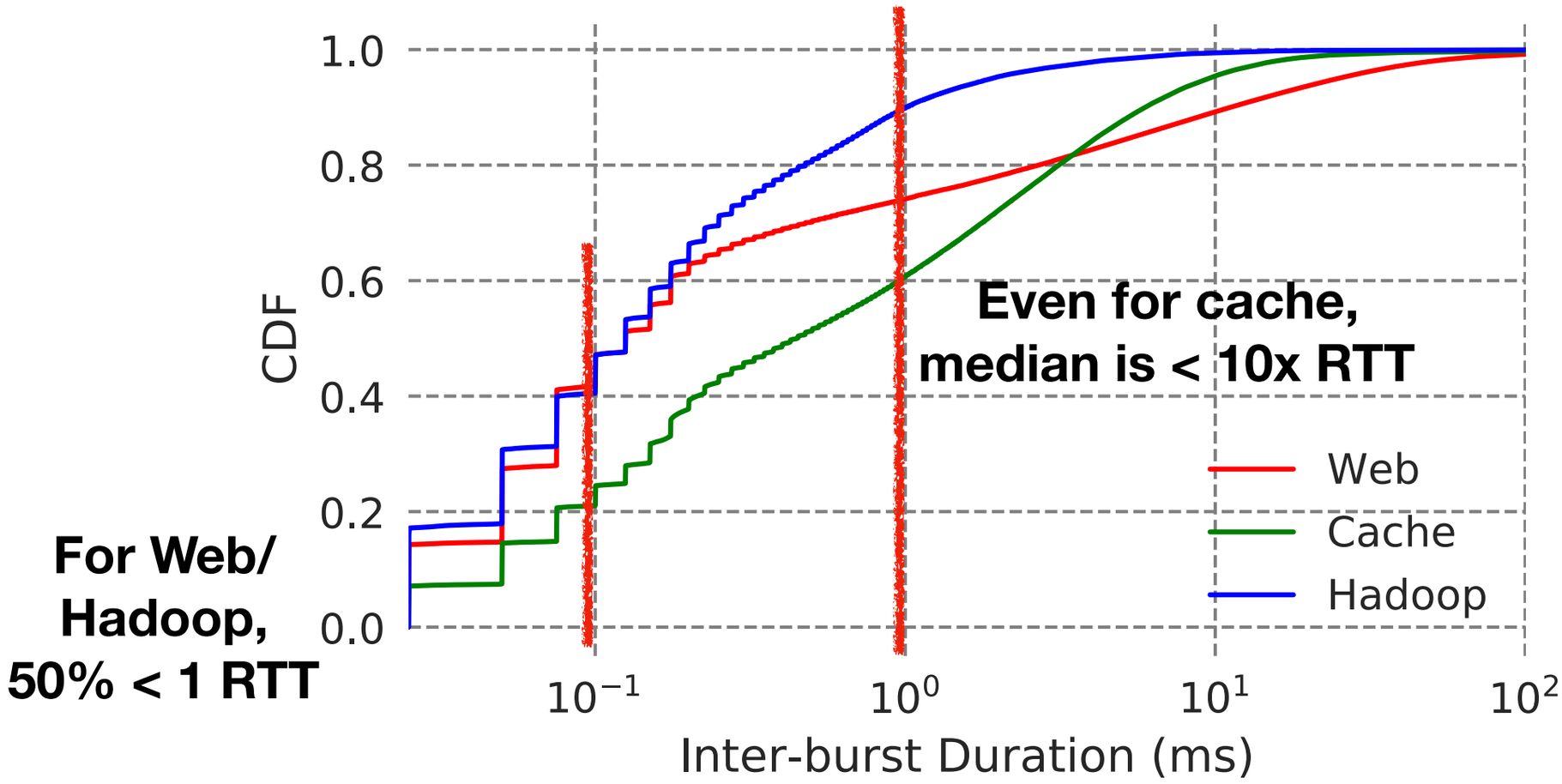
25 µs

Time between Bursts



25 µs

Time between Bursts

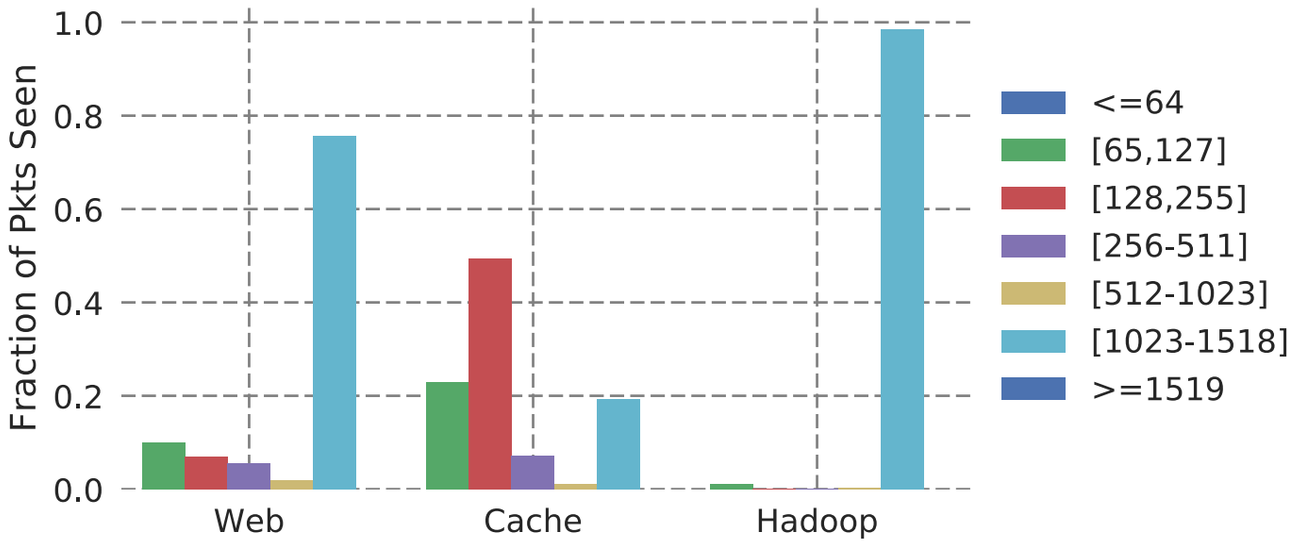


- **Some predictability: a burst is likely to be followed by another relatively soon**

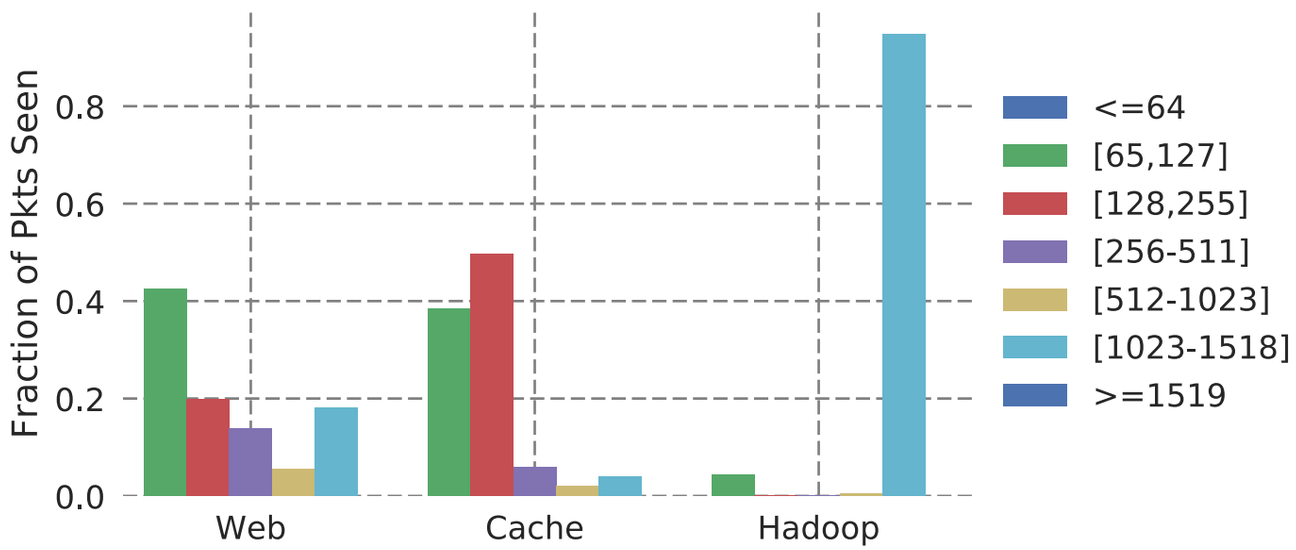
- **Potential for re-balance between bursts**

25 µs

Packet Size Distribution



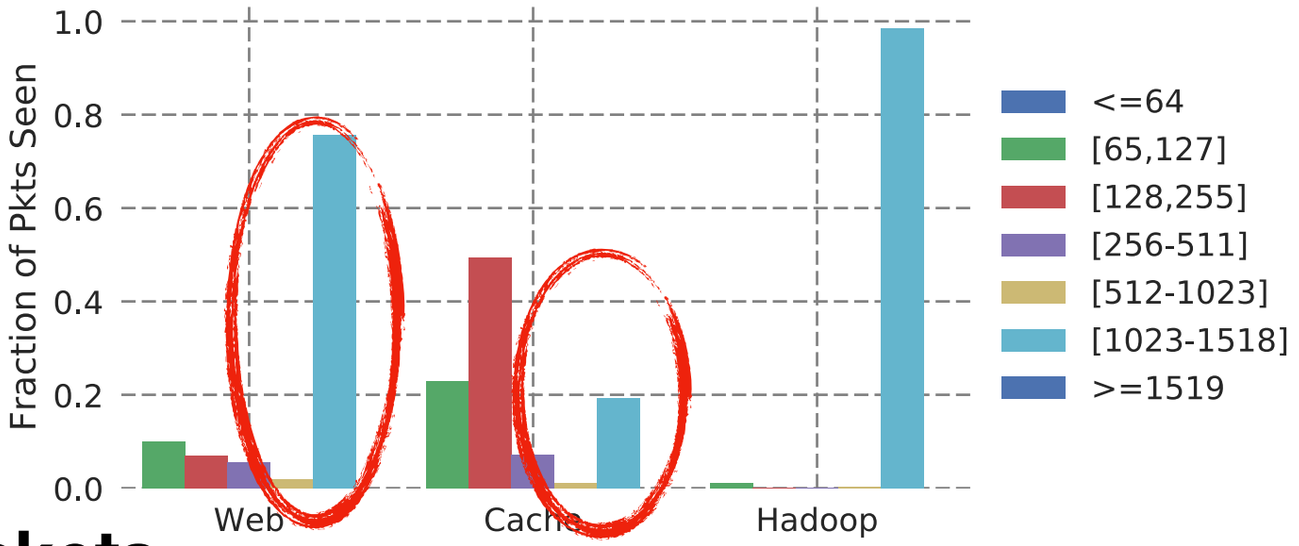
Inside Burst



Outside Burst

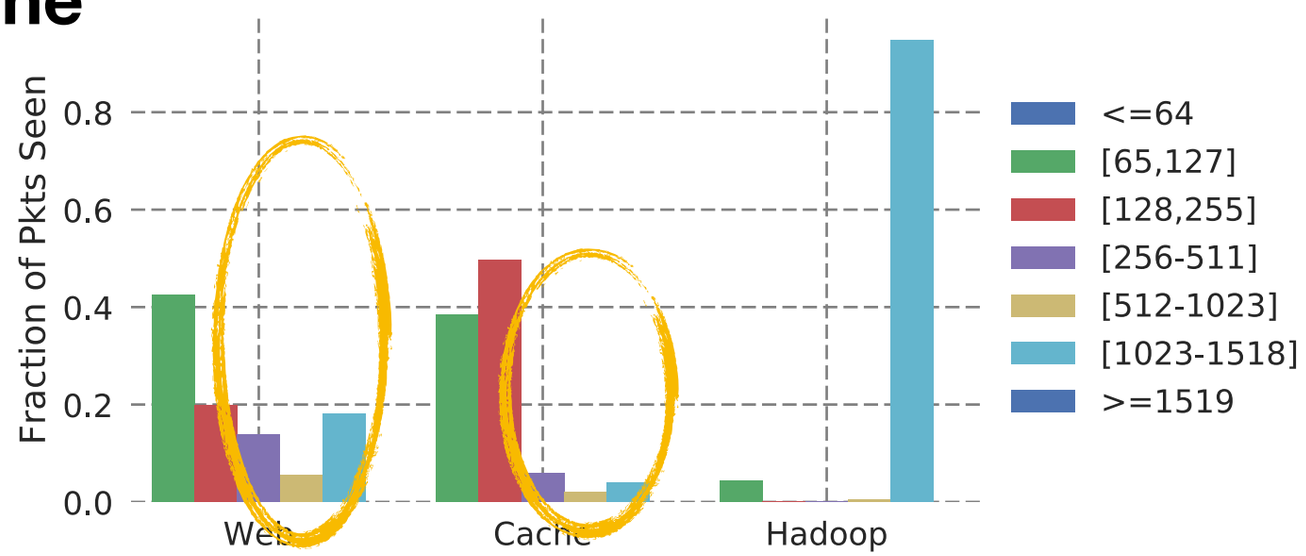
100 µs

Packet Size Distribution



Inside Burst

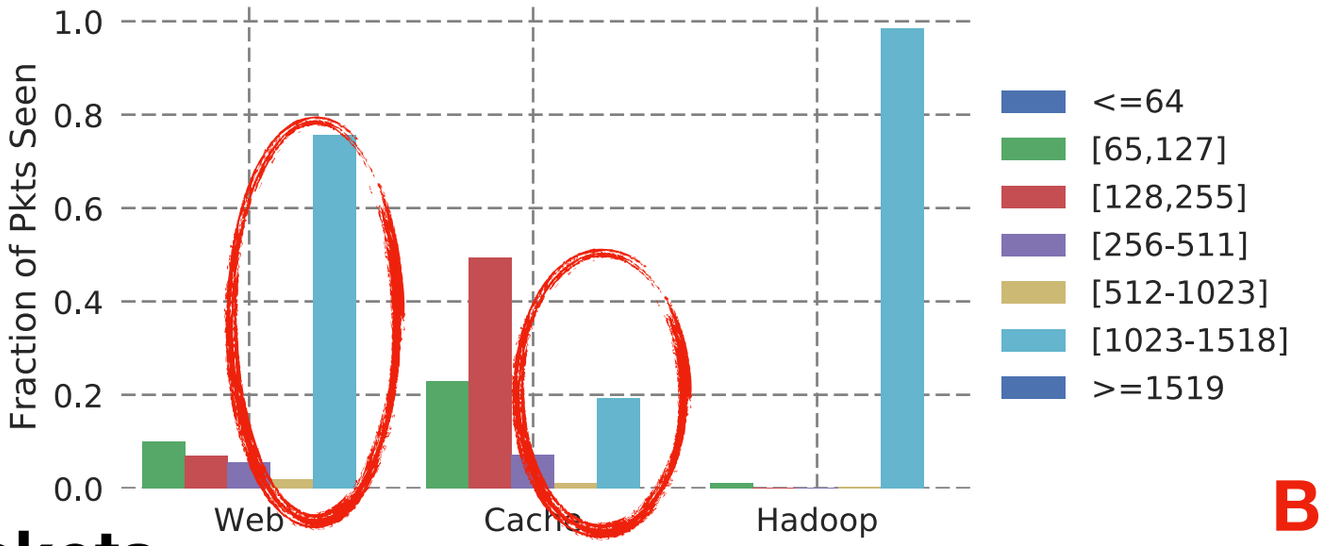
Bigger packets inside bursts for Web/Cache



Outside Burst

100 µs

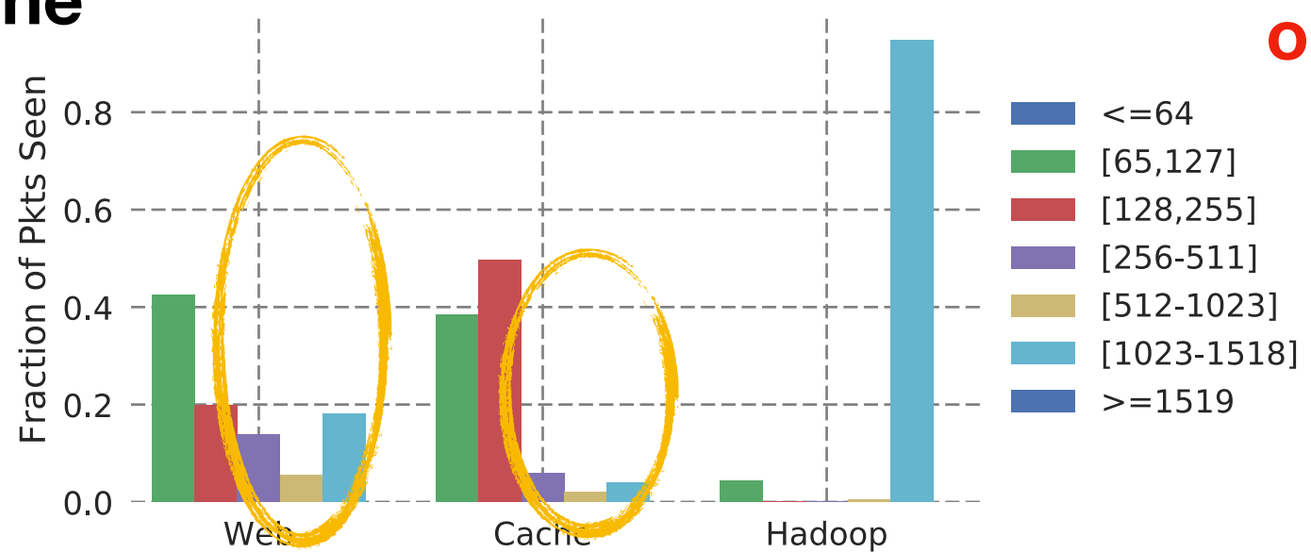
Packet Size Distribution



Inside Burst

Bigger packets inside bursts for Web/Cache

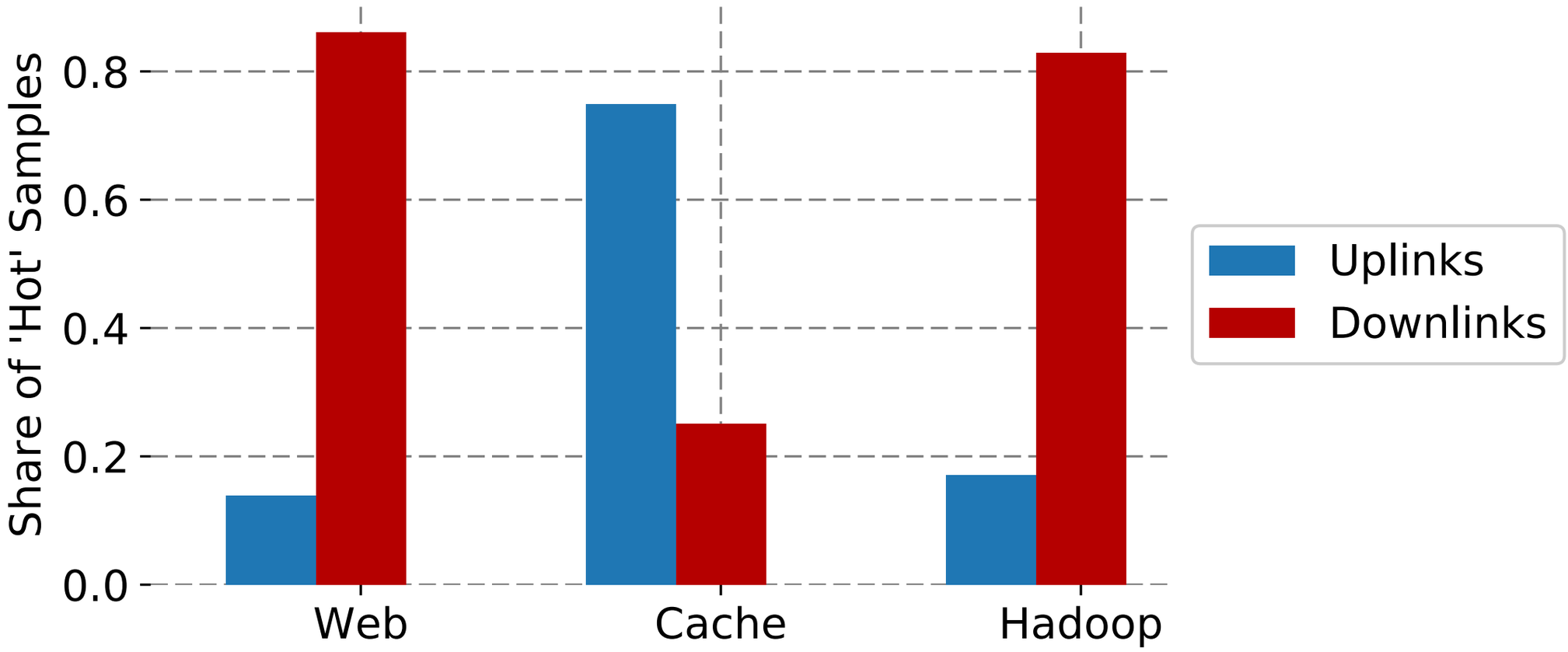
Burst are correlated with app-level behaviors (e.g. sending bigger responses or scatter-gather/incast)



Outside Burst

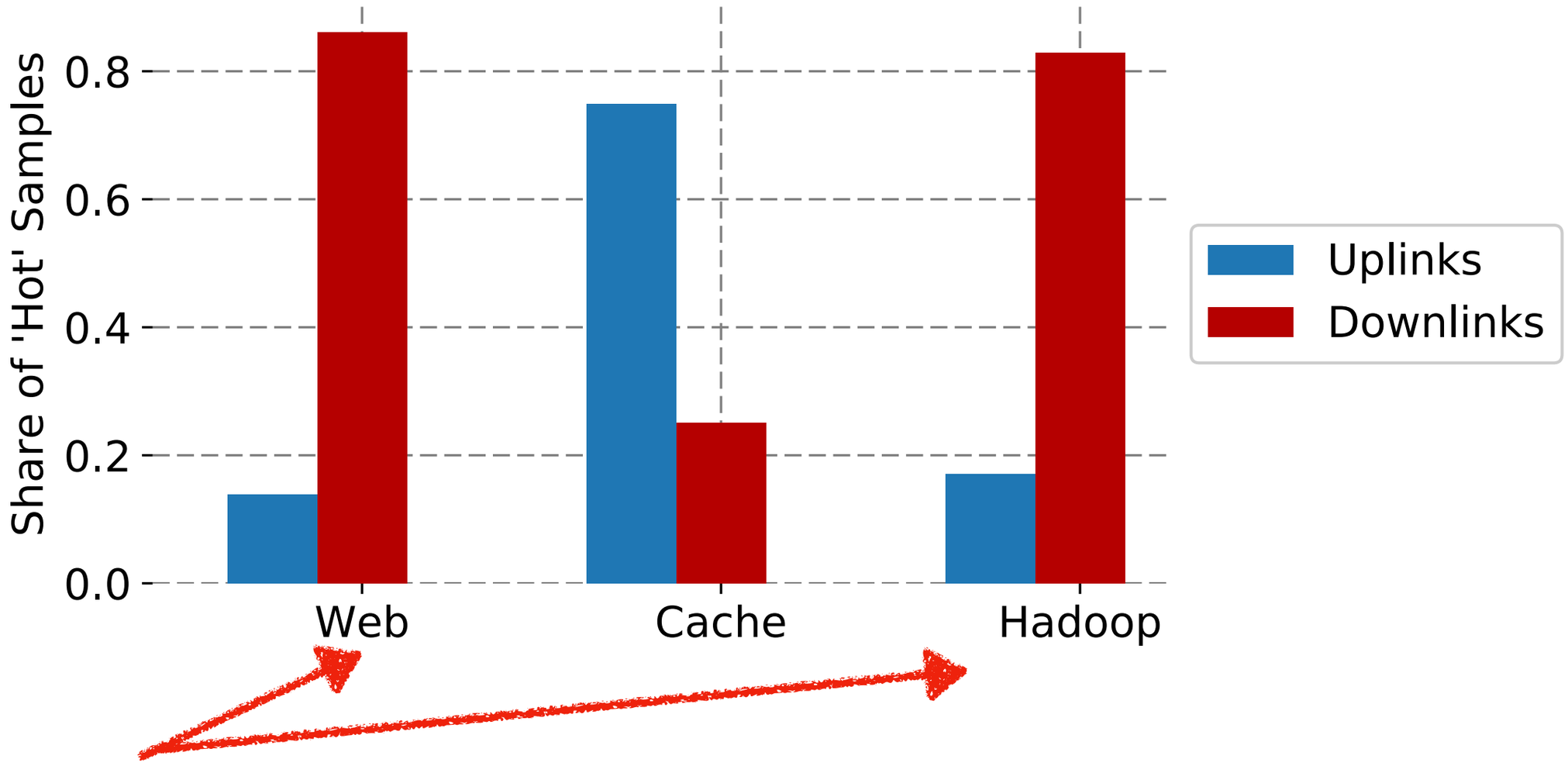
100 µs

Directionality of Bursts



300 µs

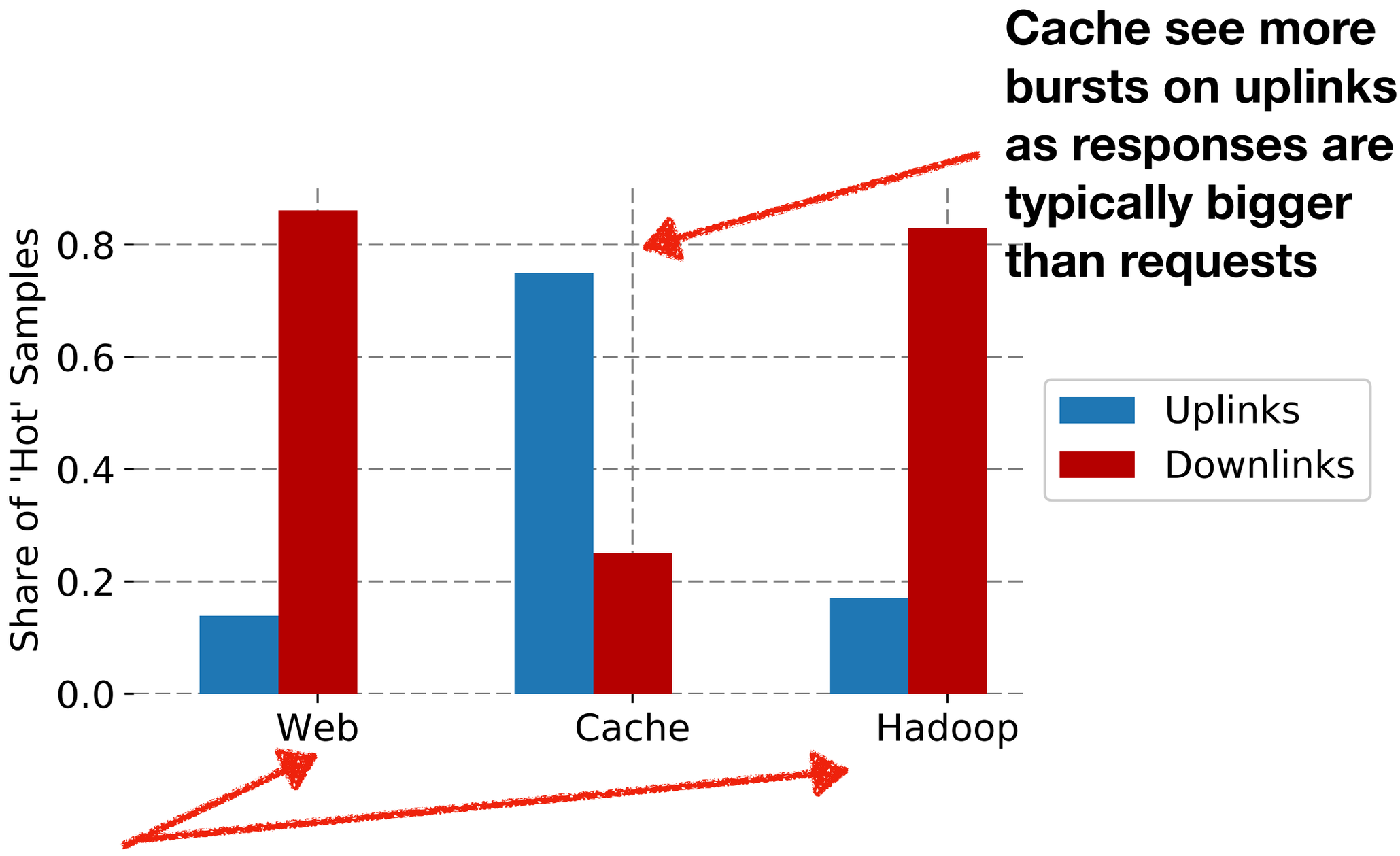
Directionality of Bursts



More bursts towards servers due to high fan-in

300 µs

Directionality of Bursts

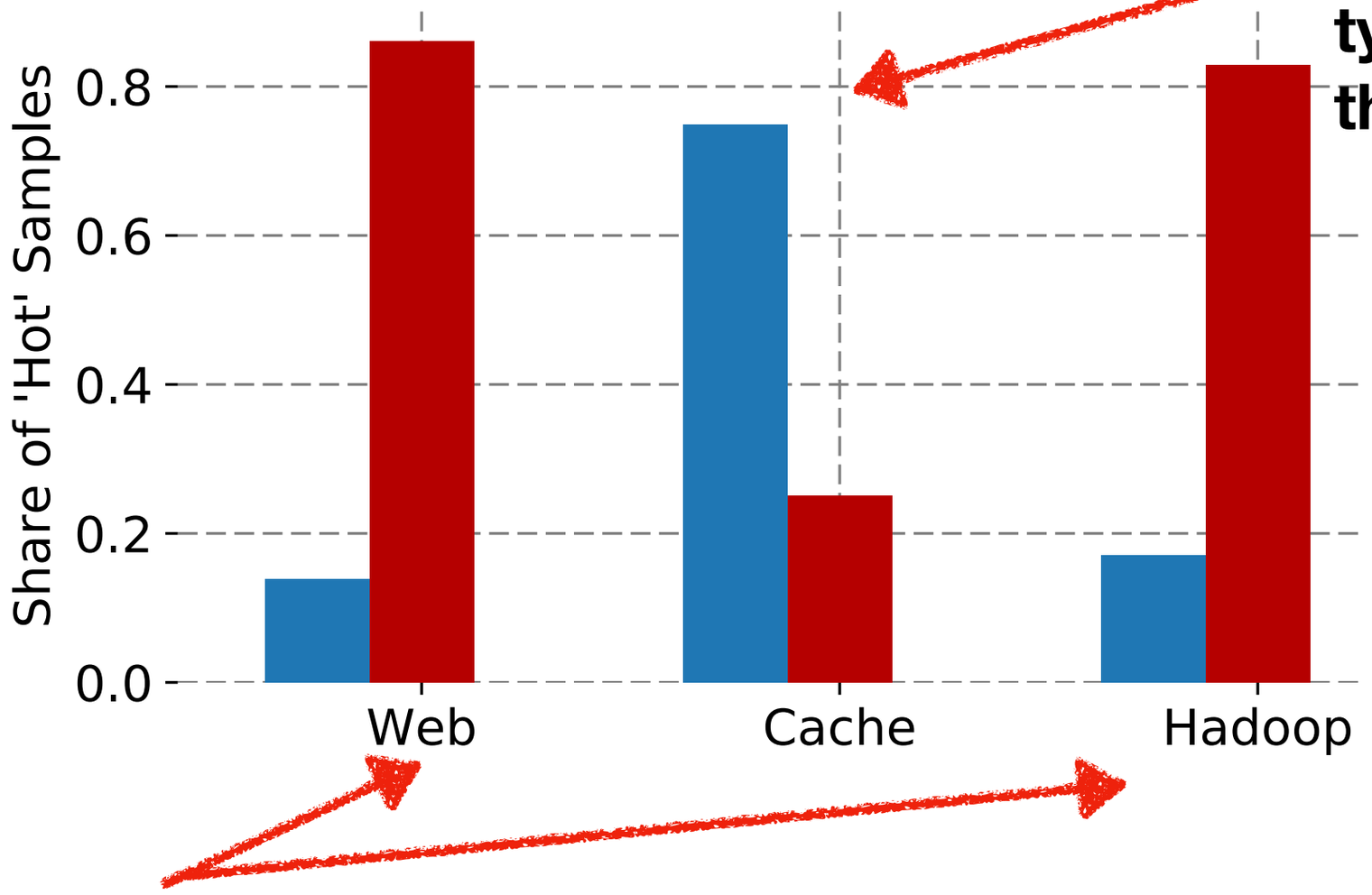


Cache see more bursts on uplinks as responses are typically bigger than requests

More bursts towards servers due to high fan-in

300 µs

Directionality of Bursts



Cache see more bursts on uplinks as responses are typically bigger than requests

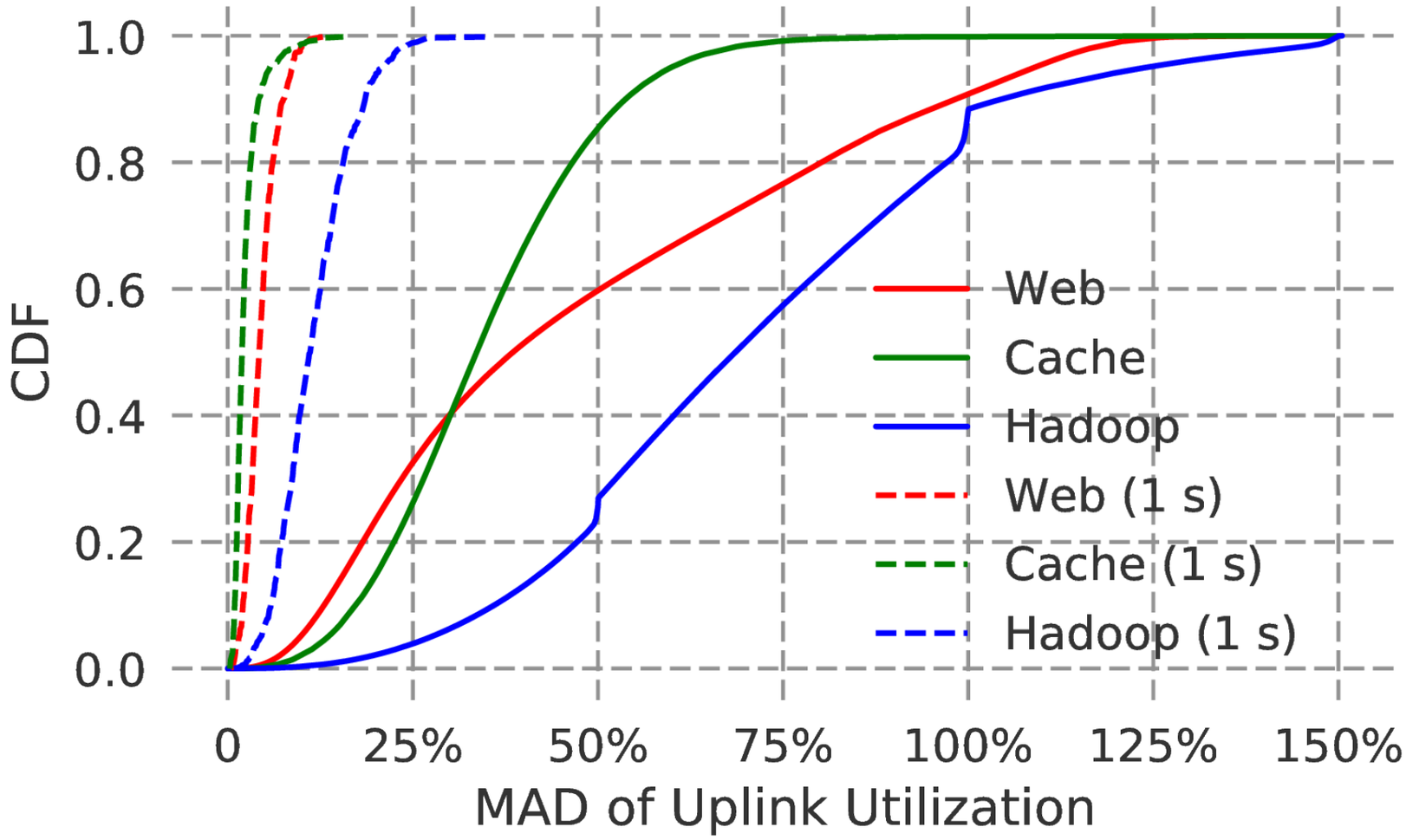
More bursts towards servers due to high fan-in

Bursts are correlated with app behaviors

300 µs

Efficacy of Network Load Balancing

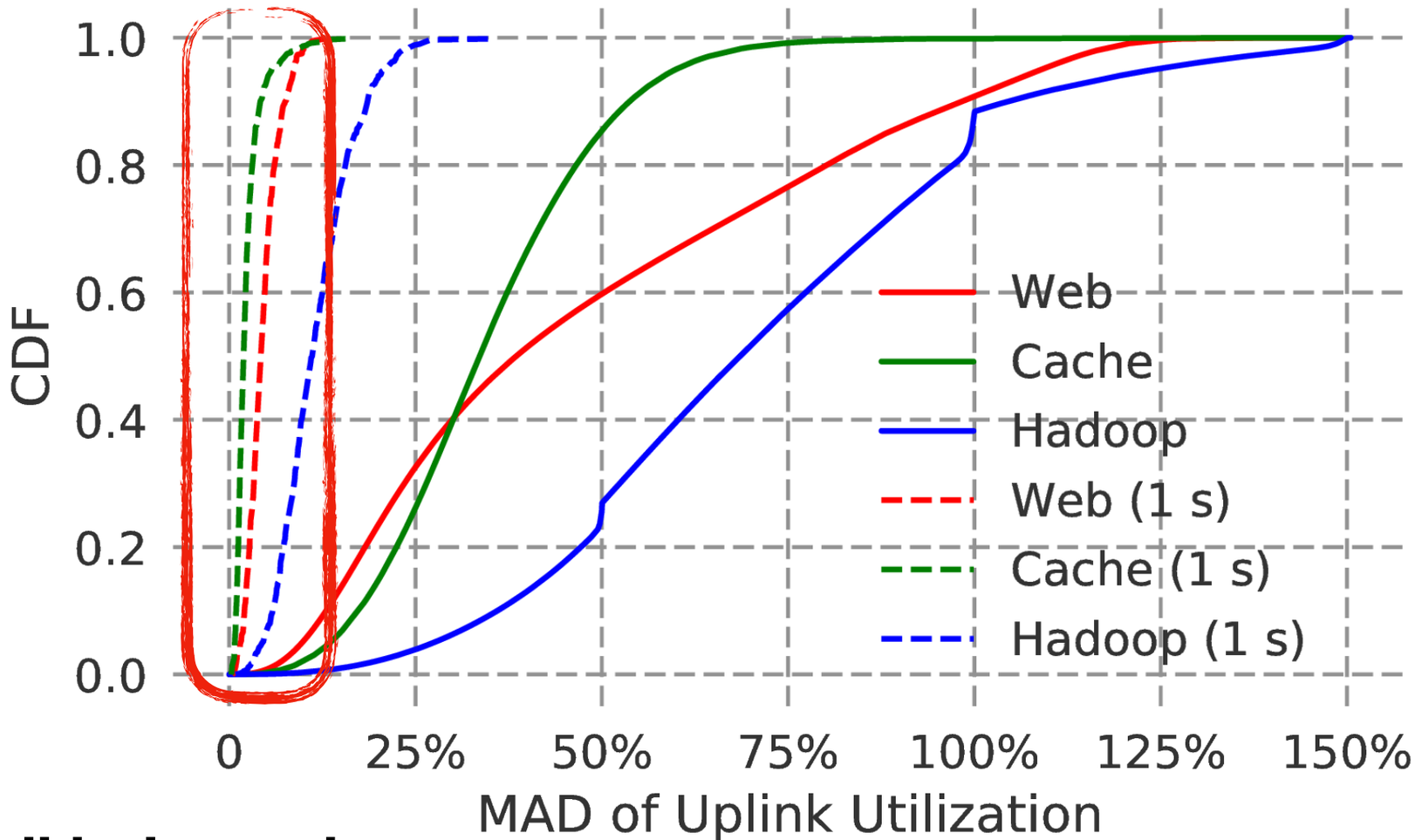
- 4 ToR Uplinks: compute mean absolute deviation (MAD) for each polling interval
- MAD = mean(|u - ū| / ū), so MAD=0 means perfect load balancing



40 µs

Efficacy of Network Load Balancing

- 4 ToR Uplinks: compute mean absolute deviation (MAD) for each polling interval
- MAD = mean(|u - ū| / ū), so MAD=0 means perfect load balancing

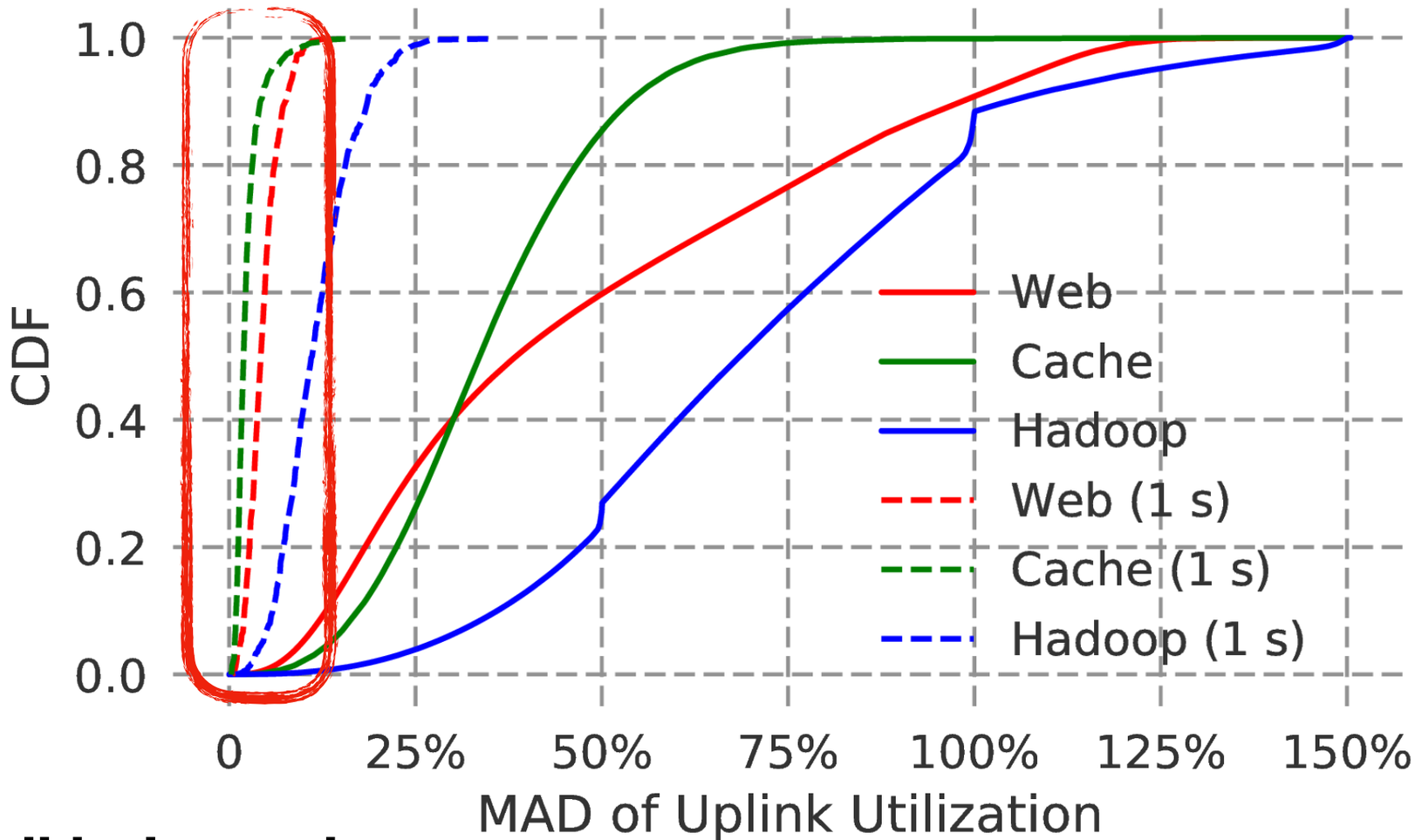


links well balanced at 1s scale

40 µs

Efficacy of Network Load Balancing

- 4 ToR Uplinks: compute mean absolute deviation (MAD) for each polling interval
- MAD = mean(|u - ū| / ū), so MAD=0 means perfect load balancing



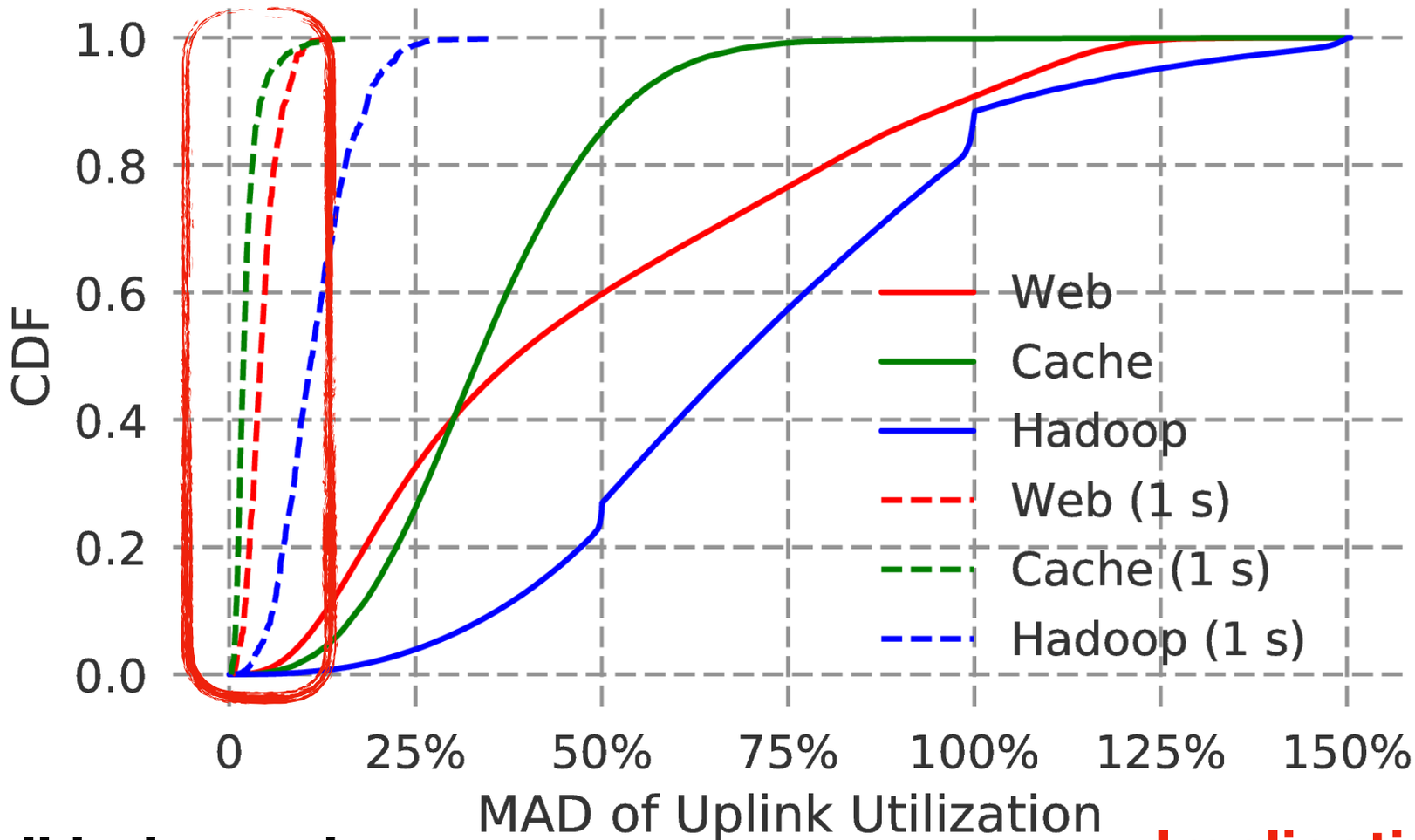
links are highly unbalanced at 40 µs scale

links well balanced at 1s scale

40 µs

Efficacy of Network Load Balancing

- 4 ToR Uplinks: compute mean absolute deviation (MAD) for each polling interval
- MAD = mean(|u - ū| / ū), so MAD=0 means perfect load balancing



links are highly unbalanced at 40 µs scale

links well balanced at 1s scale

40 µs

Implications for design of network, e.g. for low latency and loss

Conclusions

- Deployed a microsecond-scale measurement framework in production
  - Demonstrated it is possible to do high-resolution measurement on today's switches
  - Microbursts are real, short, correlated, and related to application behaviors
- Future work to correlate with end-host measurements to better understand causes for microbursts