ACMMULTIMEDIA 2019
21 - 25 October 2019, Nice, France

PiTree: Practical Implementations of ABR Algorithms Using Decision Trees

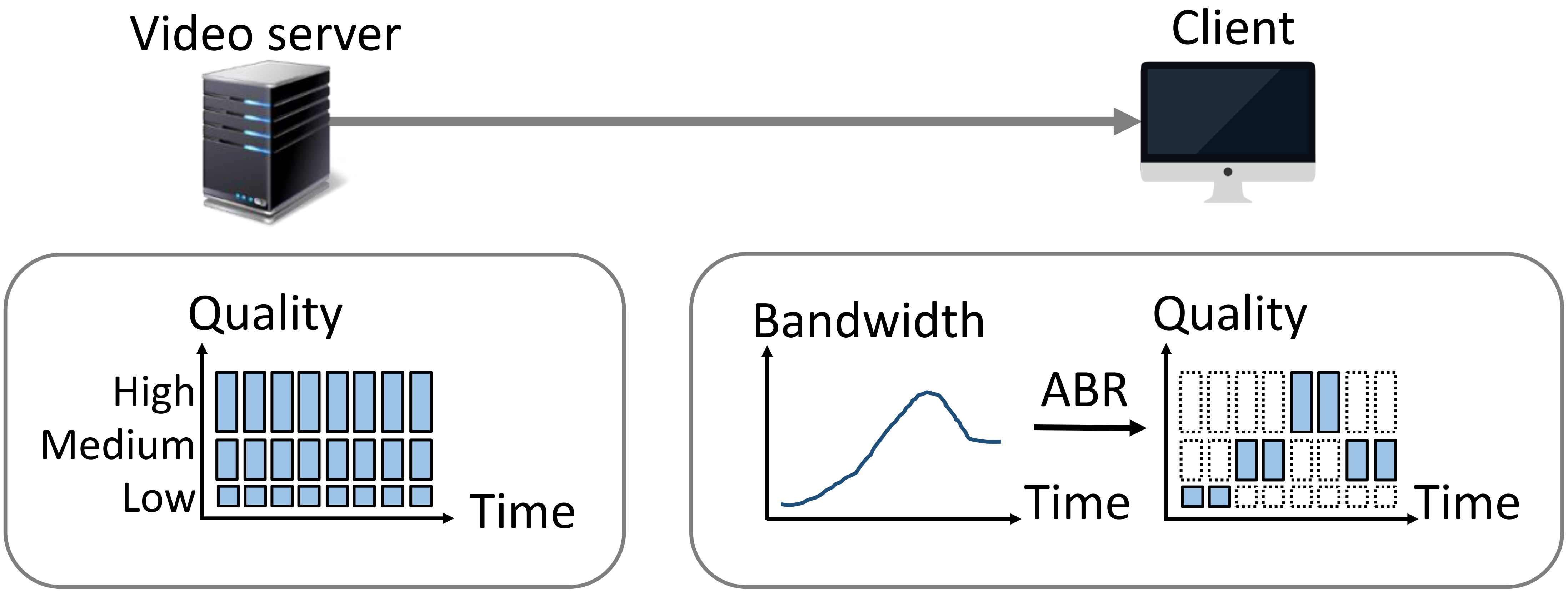
Paper # P5C-04

Zili Meng Jing Chen Yaning Guo Chen Sun Hongxin Hu Mingwei Xu

清華大學 Tsinghua University

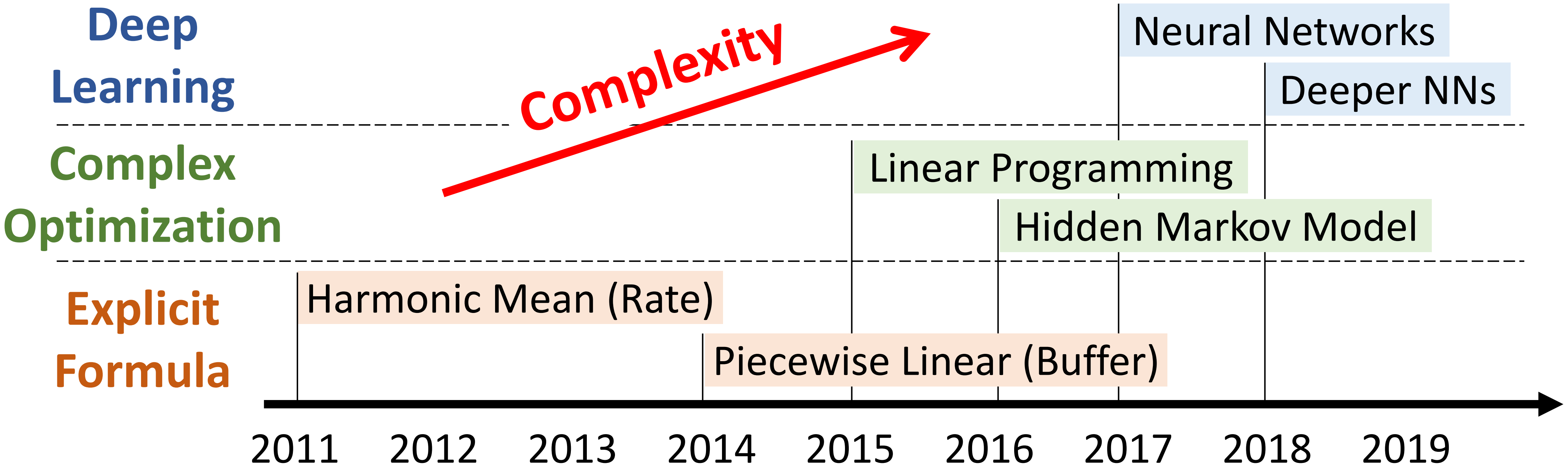
CLEMSON UNIVERSITY

Adaptive Bitrate (ABR) Algorithms



Partially borrowed from “Neural Adaptive Content-aware Internet Video Delivery” in USENIX NSDI 2018.

Adaptive Bitrate (ABR) Algorithms



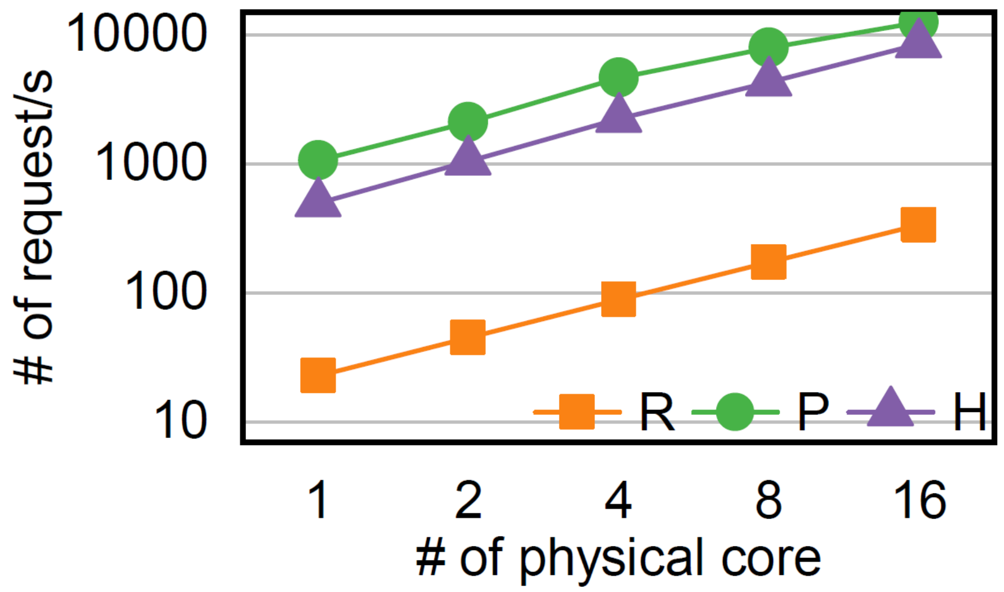
Drawbacks – Heavyweight

Client-side Implementation

- Large page size.
  - Pensieve [SIGCOMM'17] increased HTML page size by 4x.
  - Page load time is increased by ~10s.
- Long decision latency.
  - RobustMPC [SIGCOMM'15] increased decision latency to seconds.
  - Decision latency > chunk length.

Server-side Implementation

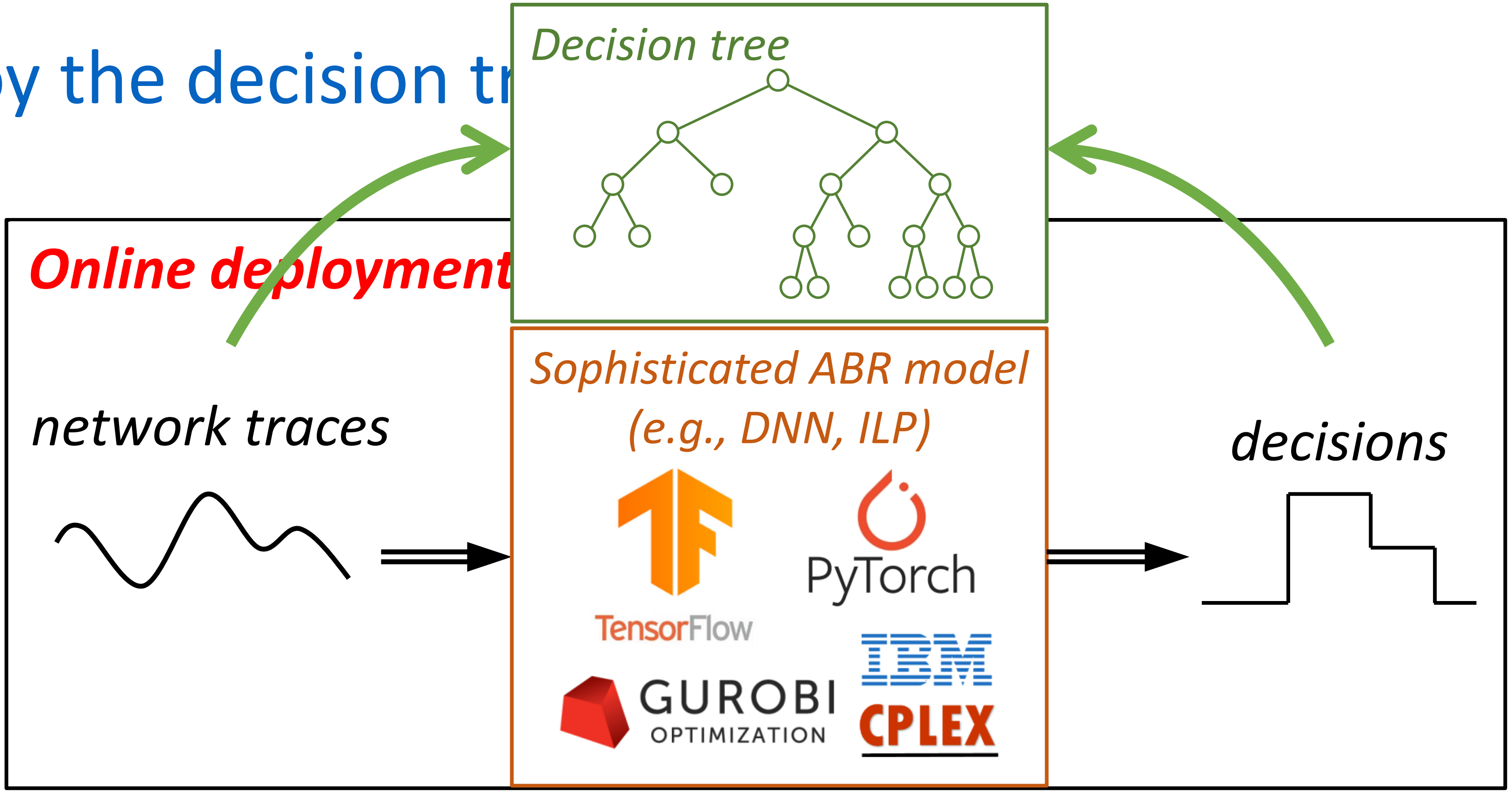
- High operating expenses.
  - Up to millions of concurrent viewers.



R = RobustMPC [SIGCOMM'15]: ILP
P = Pensieve [SIGCOMM'17]: DNN
H = HotDASH [ICNP'18]: 2xDNN

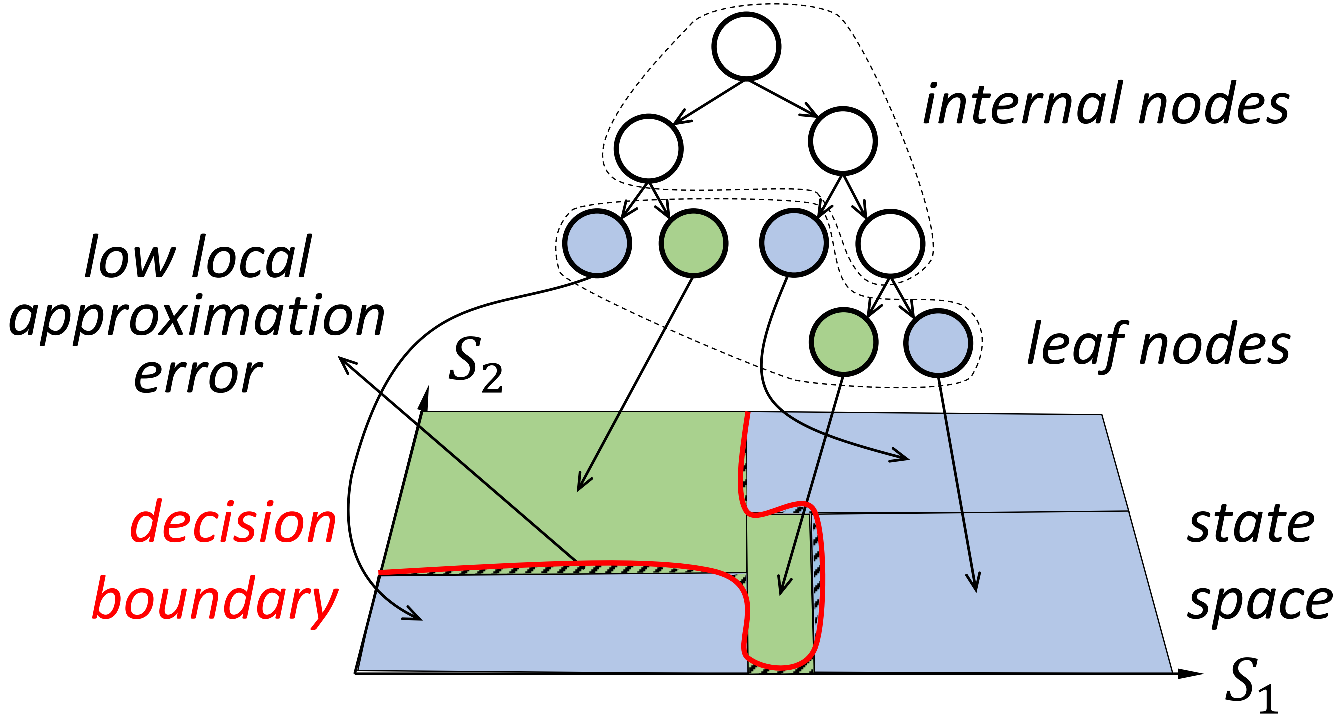
Our Contribution: PiTree

- Design & train ABR algorithms offline as usual.
- Convert the model into a decision tree.
- Deploy the decision tr



Design Choice: Why Decision Tree?

- Non-parametric and expressive.



Design Choice: Why Decision Tree?

- Non-parametric and expressive.
- Lightweight for video players.

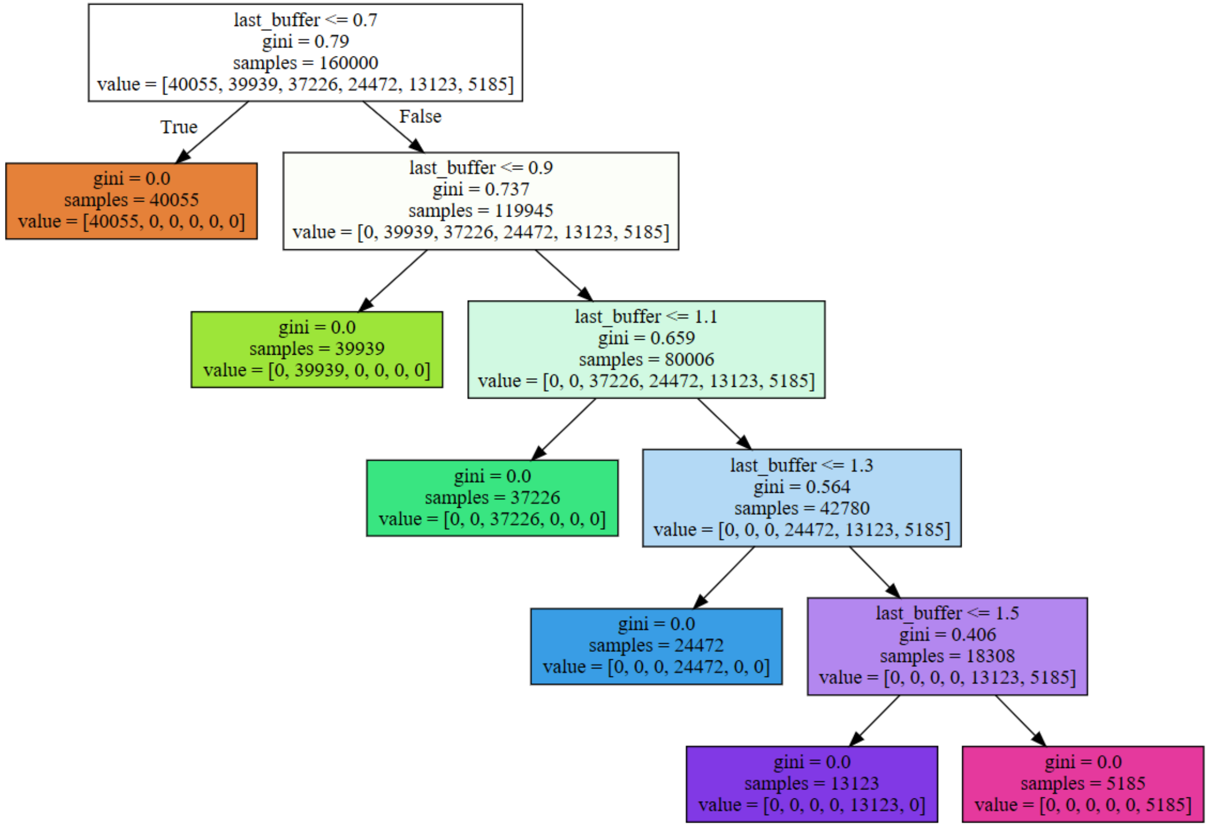
A decision tree with 100 leaf nodes:
Page size increase <1%
Decision latency <1ms

laptop

smartphone

Design Choice: Why Decision Tree?

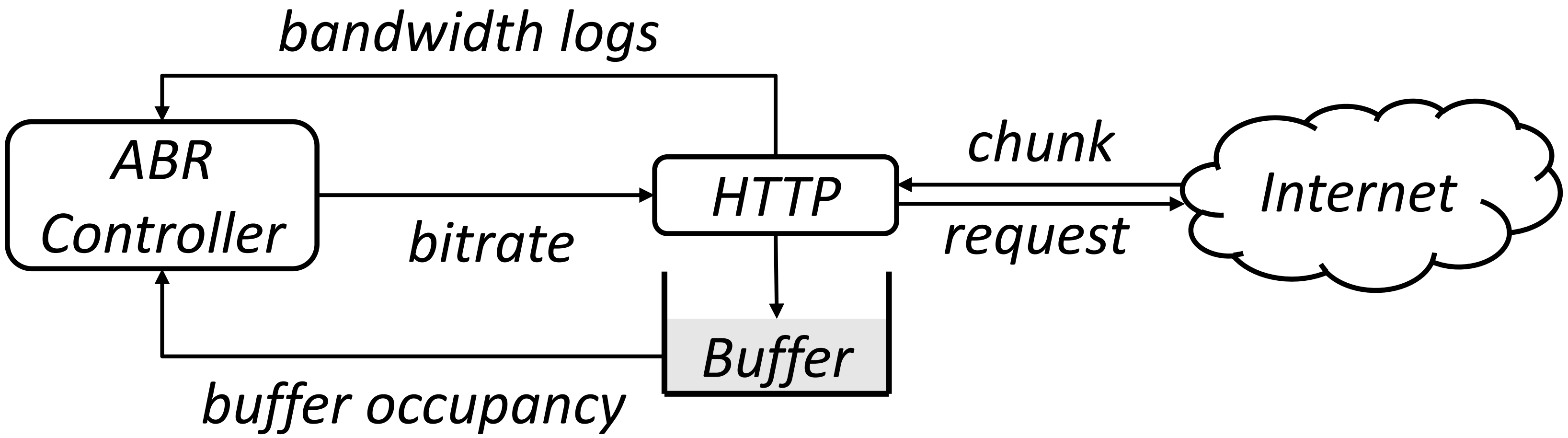
- Non-parametric and expressive.
- Lightweight for video players.
- Following the decision logic of ABR algorithms.



Decision tree of BBA [SIGCOMM'14].

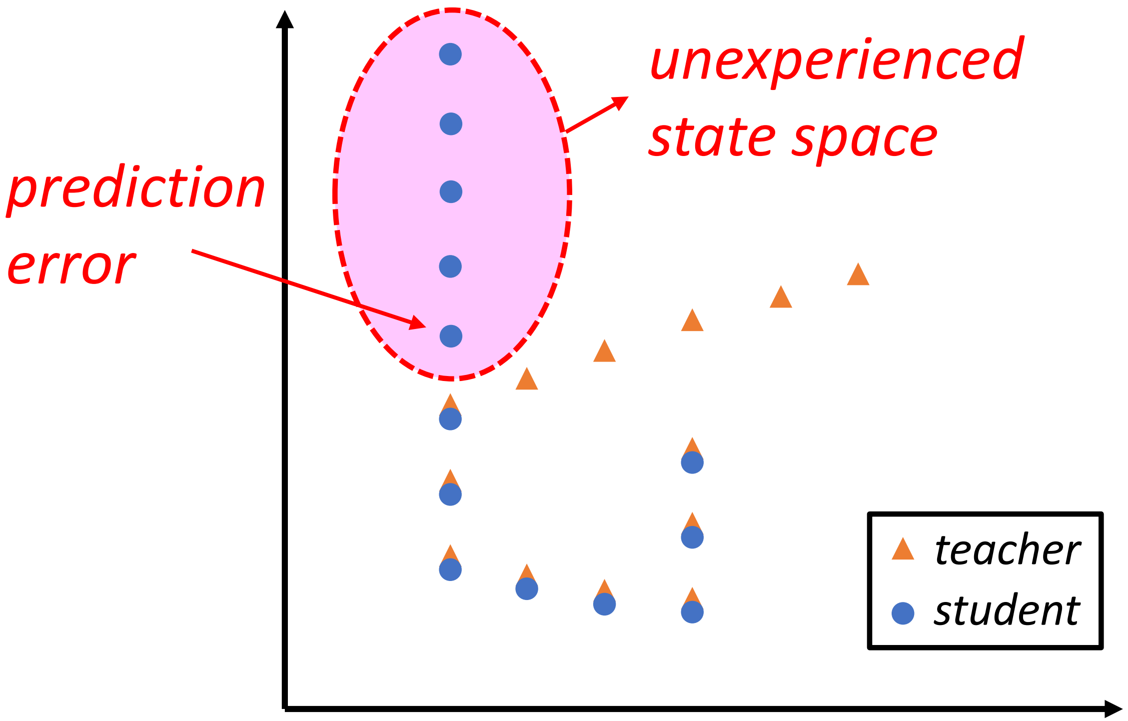
Design Challenge: Sequential Dependency

- ABR Control is a sequential decision-making process.

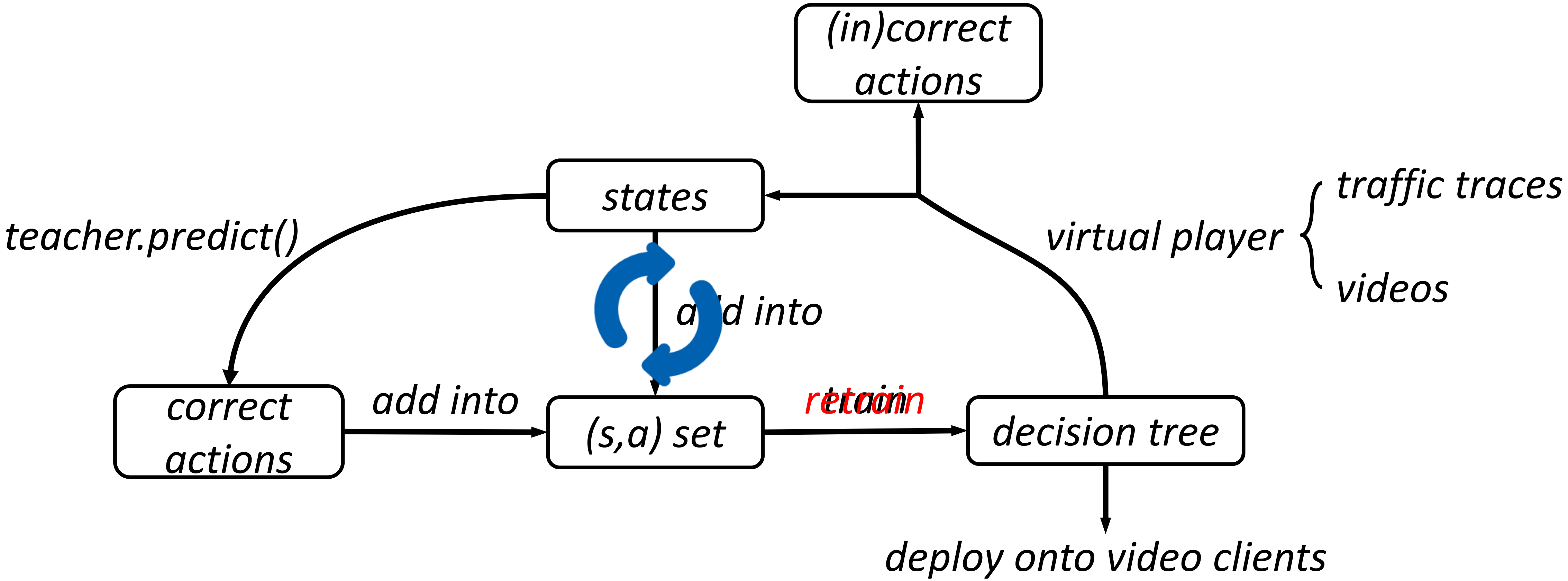


Design Challenge: Sequential Dependency

- ABR Control is a sequential decision-making process.
- One wrong prediction may drive the student off teacher’s trajectory.



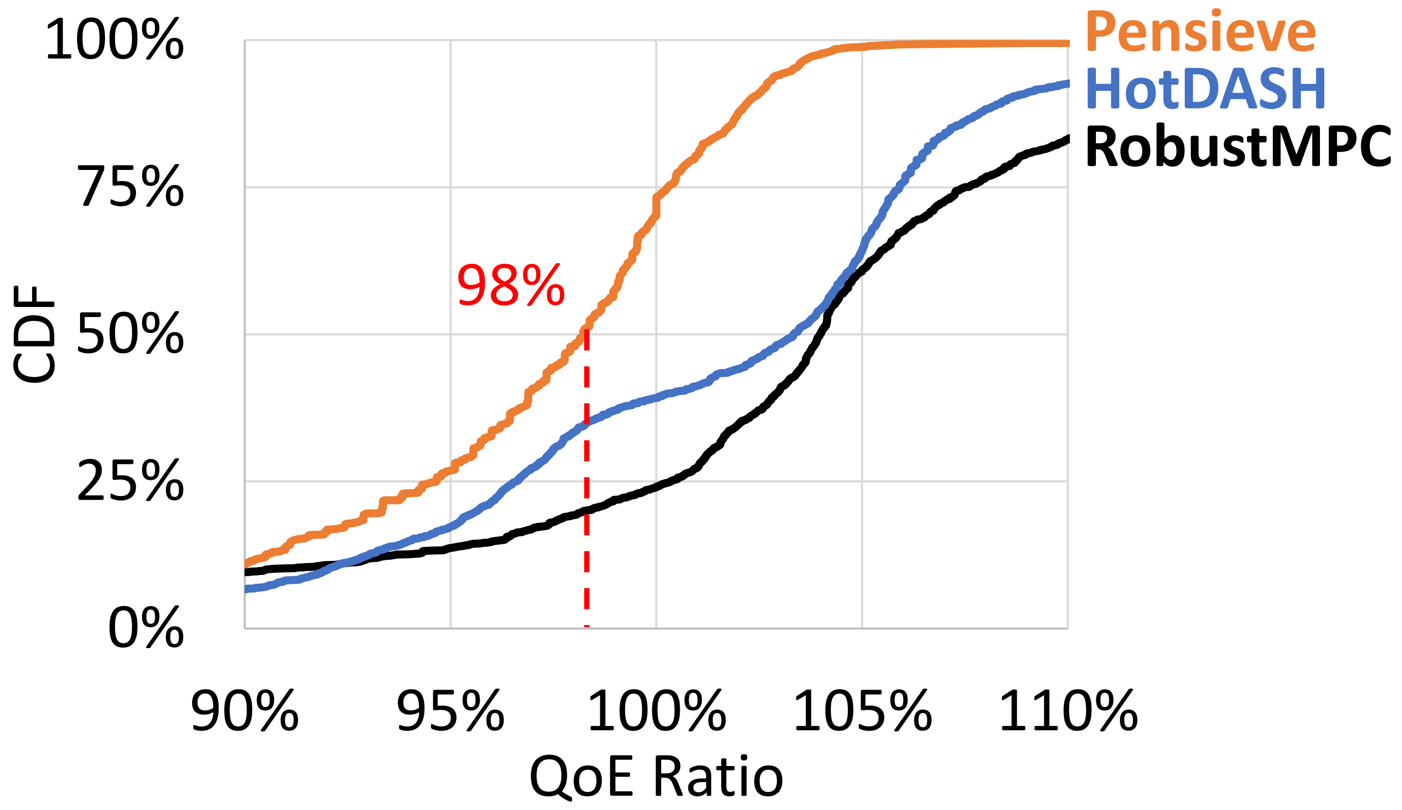
PiTree: Imitation Learning – Follow the Leader



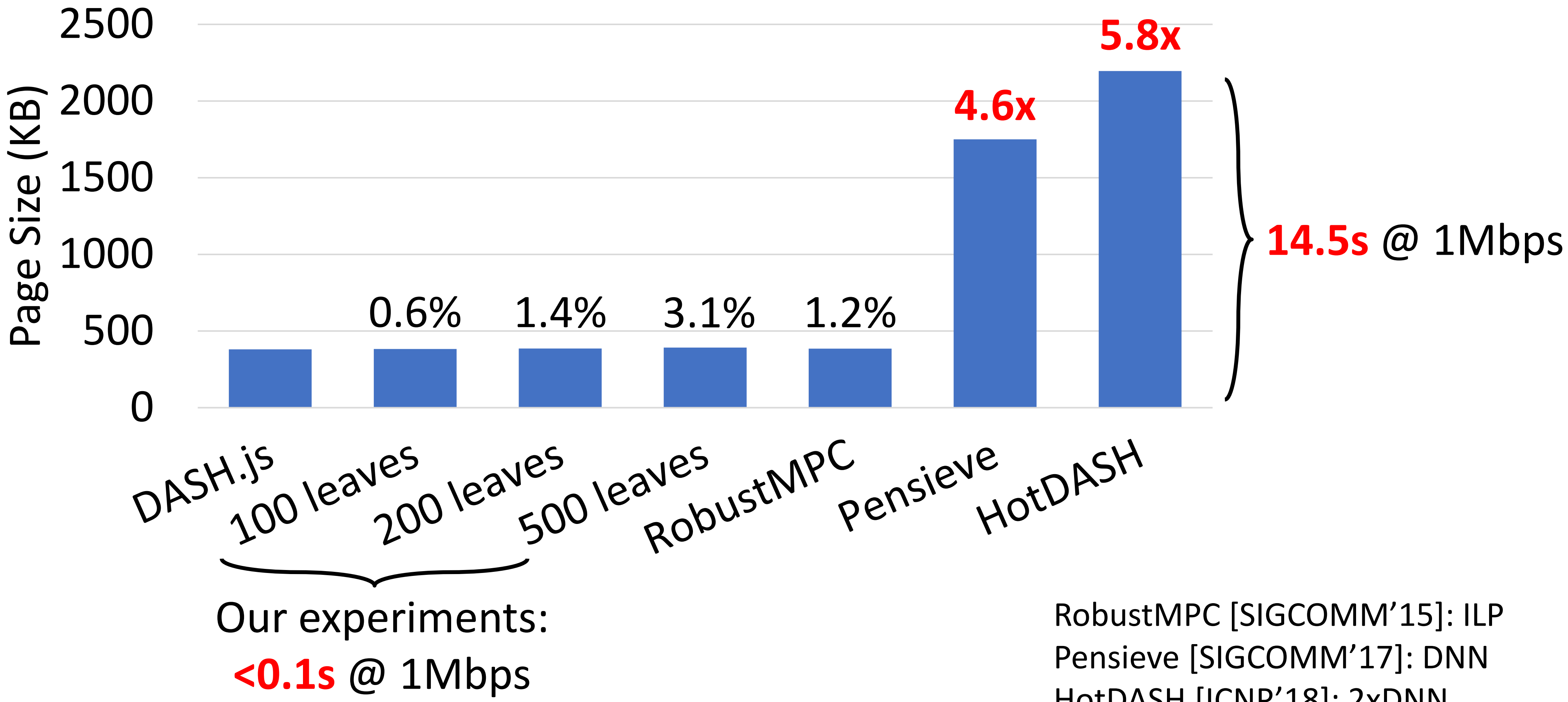
For design details and theoretical analysis, please refer to our paper.

Evaluation – Quality of Experience (QoE) Ratio

- Summary of experiments
  - 3 QoE metrics.
  - 3 sets of bandwidth traces.
  - 3 ABR algorithms.
- $\text{QoE ratio} = \frac{\text{QoE}_{\text{PiTree}}}{\text{QoE}_{\text{Original}}}$
- Average QoE ratio > 97%.
- Median QoE ratio > 98%.
- Details in the paper.



Evaluation – Page Size

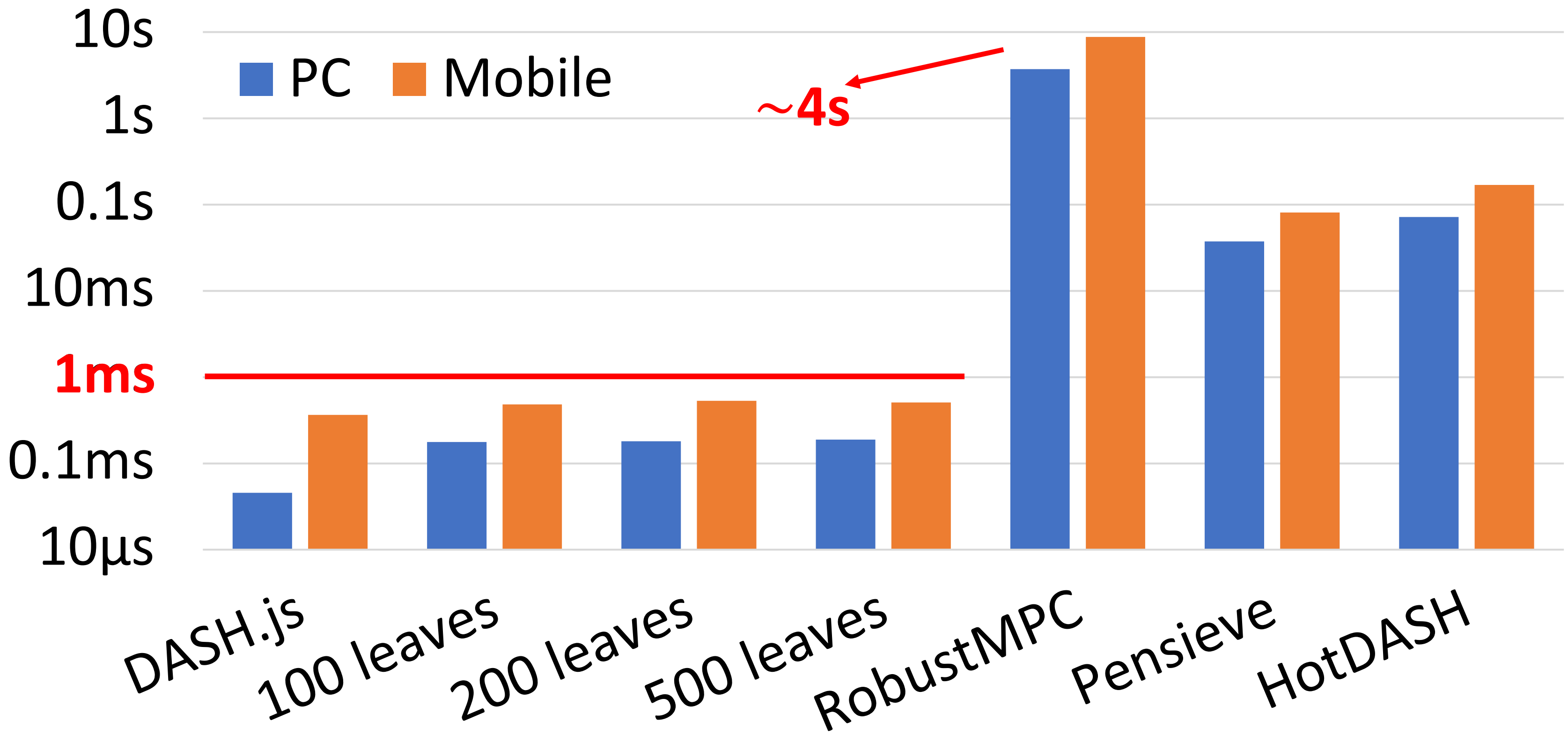


RobustMPC [SIGCOMM'15]: ILP
Pensieve [SIGCOMM'17]: DNN
HotDASH [ICNP'18]: 2xDNN

Evaluation – Latency

Intel Core i7-8550

Qualcomm Snapdragon 821



RobustMPC [SIGCOMM'15]: ILP
Pensieve [SIGCOMM'17]: DNN
HotDASH [ICNP'18]: 2xDNN

Takeaways

- Current ABR algorithms are increasingly heavyweight.
- PiTree convert complex ABR algorithms to decision trees to deploy them in a lightweight way.
  - Use imitation learning to address the action dependency.
- PiTree can significantly reduce the algorithm overhead with negligible QoE loss.
  - Page size reduced by up to 5x, decision latency reduced by up to 1000x.

Thank you!
Questions and comments?

Try your ABR algorithms with PiTree!

https://transys.io/pitree

zilim@ieee.org