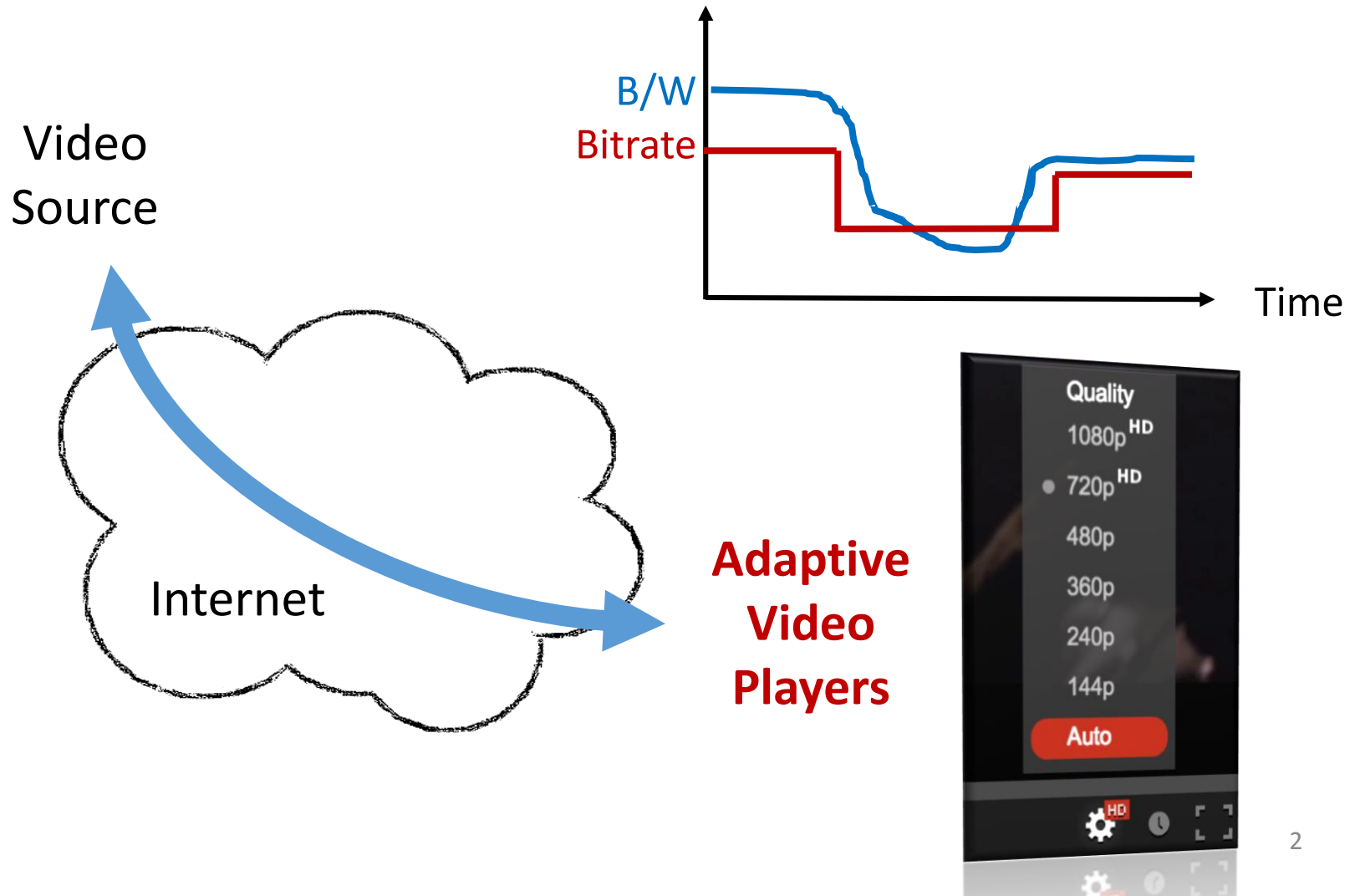


Toward a Principled Framework to Design Dynamic Adaptive Streaming Algorithms over HTTP

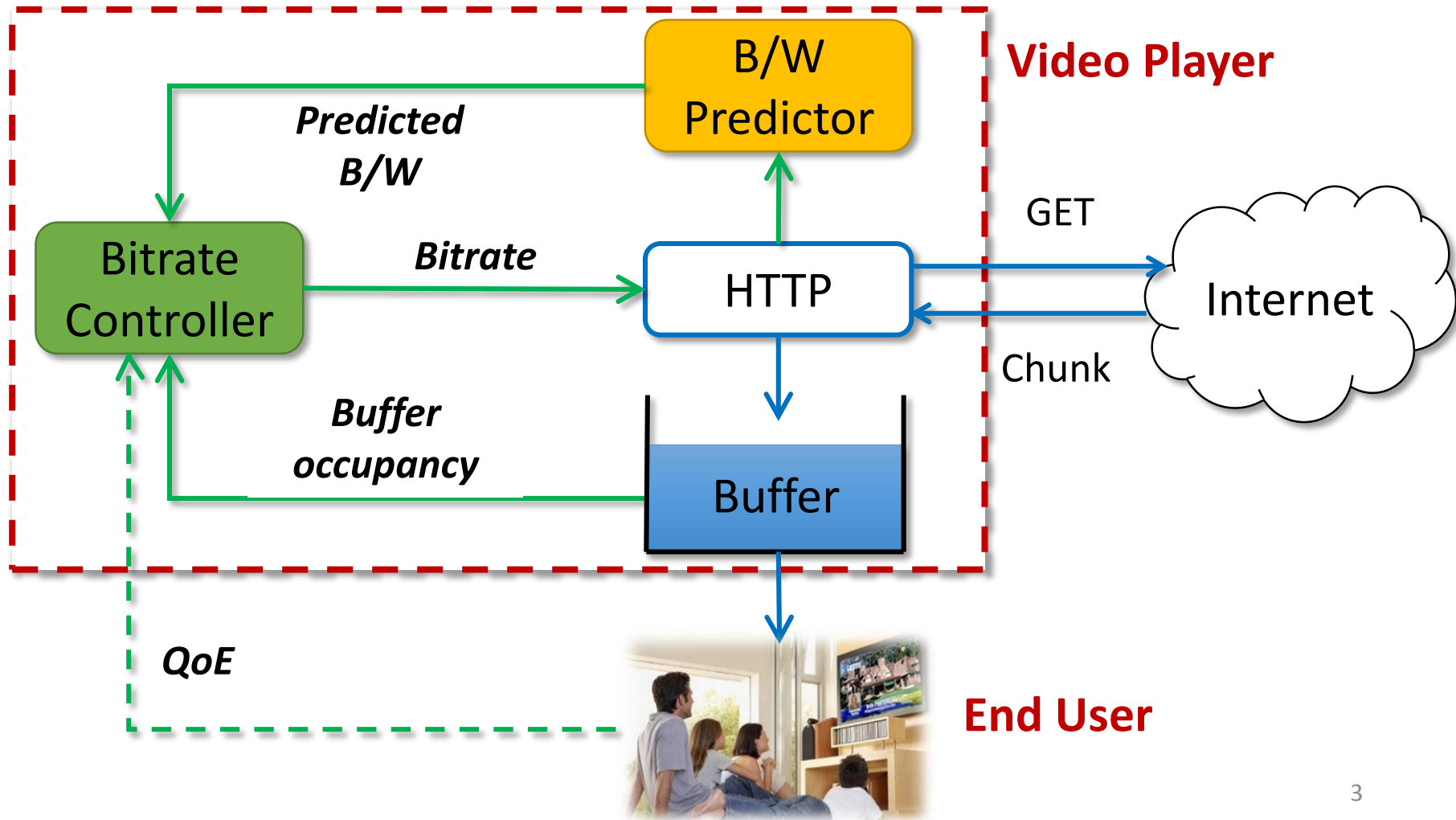
Xiaoqi Yin, Vyas Sekar, Bruno Sinopoli

Carnegie Mellon

Design Dynamic Adaptive Streaming (DASH) algorithms is critical for better QoE



Video player model



Video player adaptation is hard

- Hard to predict bandwidth
 - It is a stochastic variable difficult to estimate
- Interaction with TCP
 - Makes bandwidth estimation even harder
- Various factors can impact QoE
 - Often in conflict, e.g., high quality vs. few stalls
- Discrete feedback and control
 - Discrete bitrate levels, change bitrate only in discrete time

Plenty of algorithms, little clarity

- 50+ papers in the past 5 years

Rate-based



“Match bitrate with bandwidth”

Buffer-based

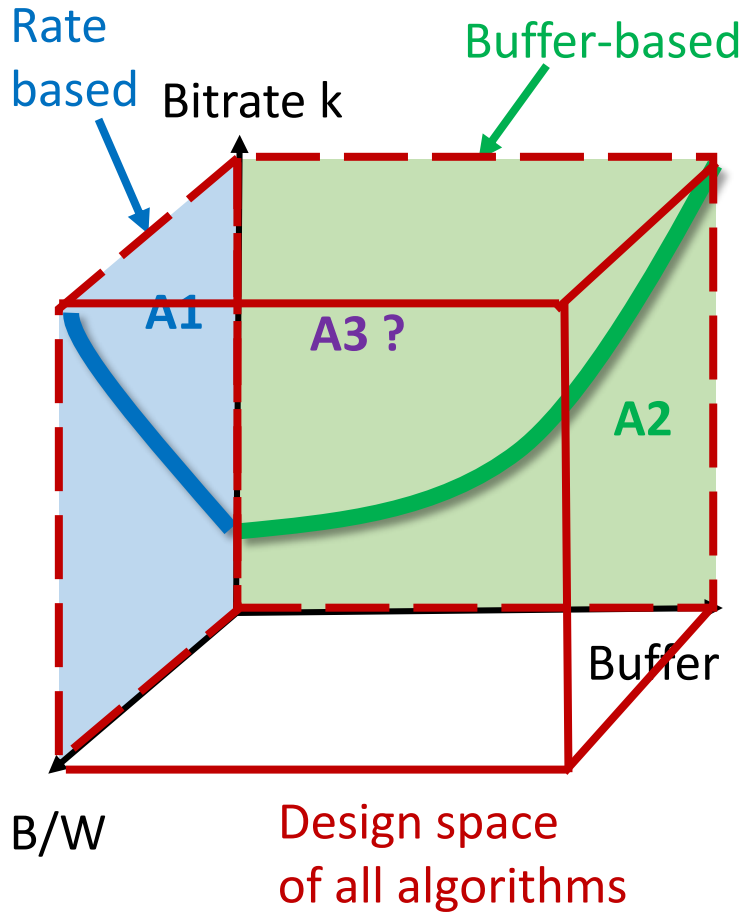


“Control buffer to certain level”



We need a systematic framework!

Several point solutions!



Objective they (should) optimize?

i.e., Buffering vs. switching vs. bitrate

Sensitivity to operating regimes

e.g., When is A1 better than A2?

Are they optimal?

→ Broader design space is possible

Stochastic optimal control framework

Choose optimal bitrate for all chunks in a video

$$R_1, R_2, \dots, R_K$$

Formally
Defined

To maximize QoE:

$q(\text{avg bitrate, bitrate switches, rebuffer time})$

Sensitivity
To Operating
Parameters

Subject to:

- Buffer occupancy dynamics:

$$\text{Buf}_{k+1} = g(\text{Buf}_k, B/W, R_k)$$

- Available bandwidth

Broader
Design Space

Online controller design: $R_k = f(\text{Buf}_k, \text{Predicted B/W})$

Known precisely

Predicted with error⁷

Algorithm via Model Predictive Control (MPC)

Choose optimal bitrate for all chunks in a video

$$R_1, R_2, \dots, R_K$$

To maximize QoE:

$q(\text{avg bitrate, bitrate switches, rebuffer time})$

Subject to:

- Buffer occupancy dynamics:*

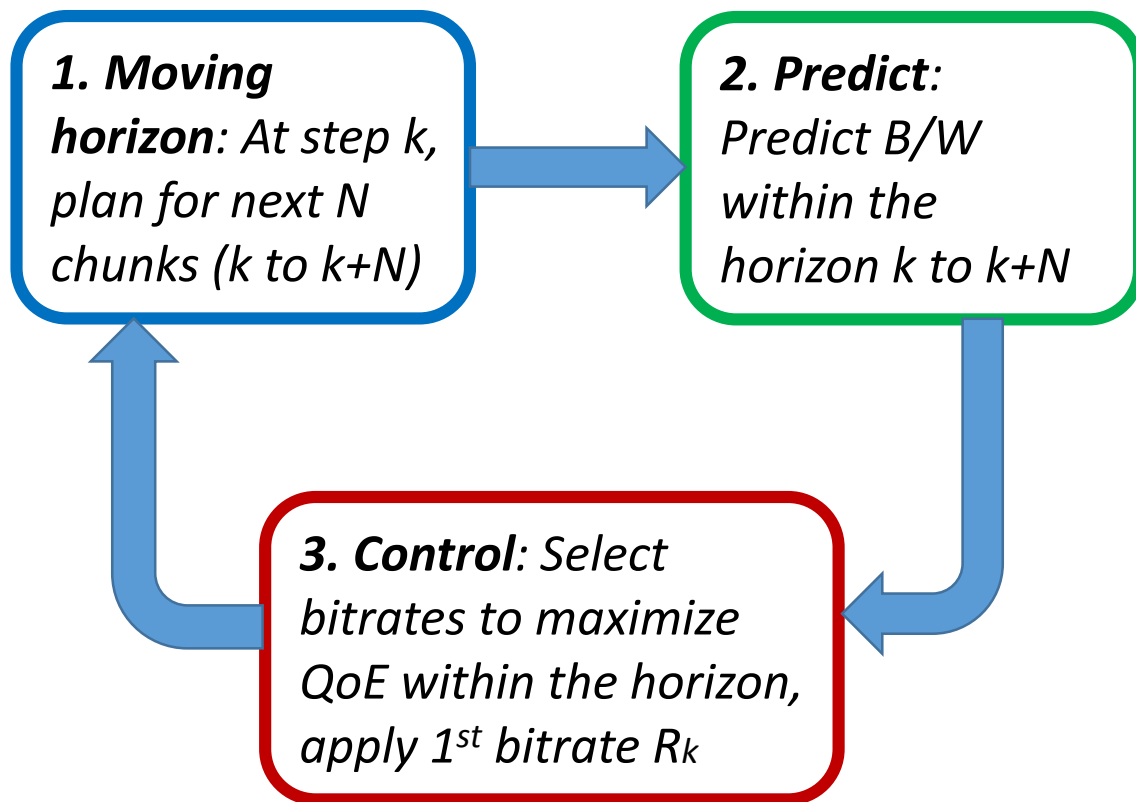
$$Buf_{k+1} = g(Buf_k, B/W, R_k)$$

- Available bandwidth*

Online controller design: $R_k = f(Buf_k, \text{Predicted B/W})$

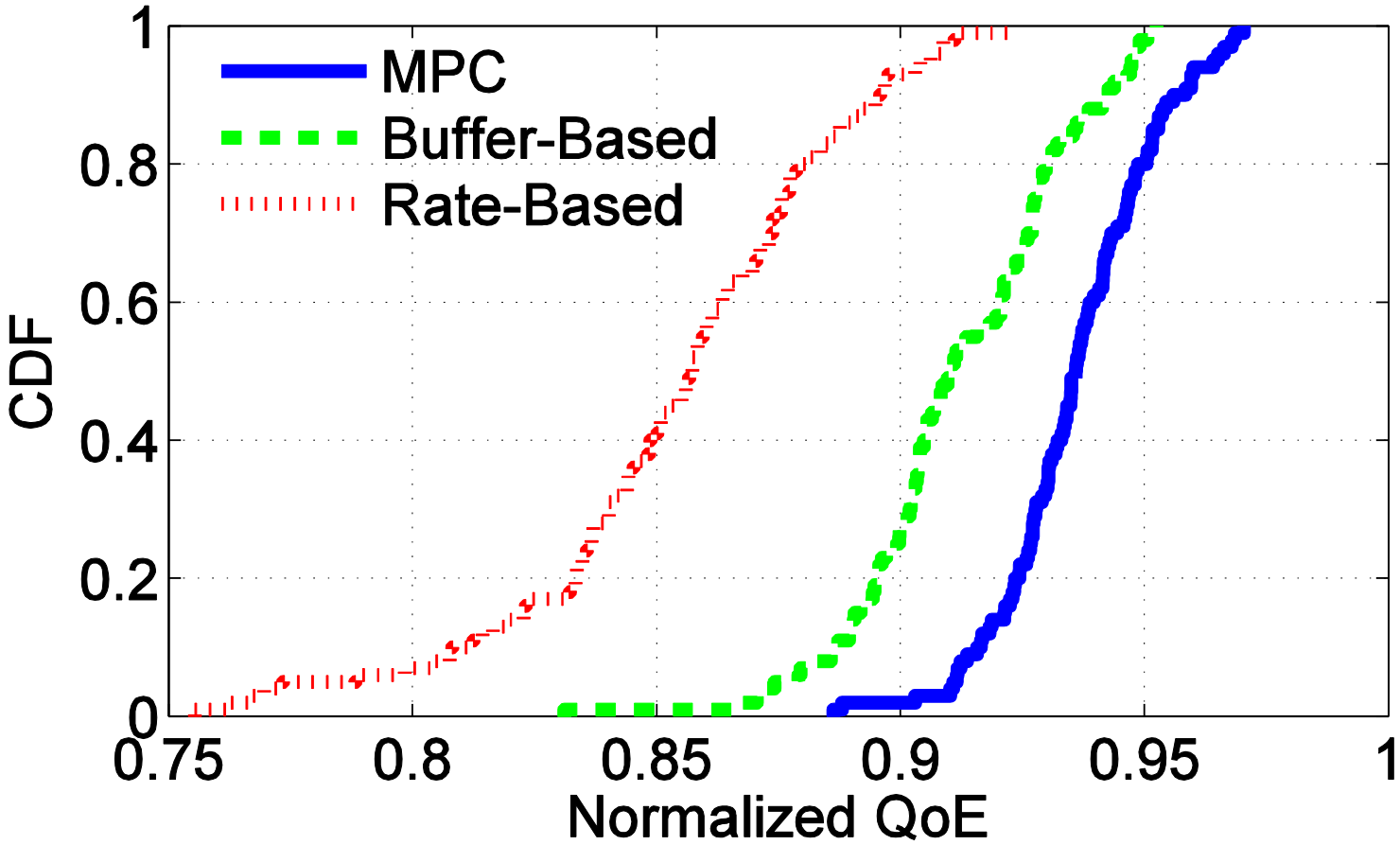
Known precisely  *Predicted with error*  ⁸

Model predictive control

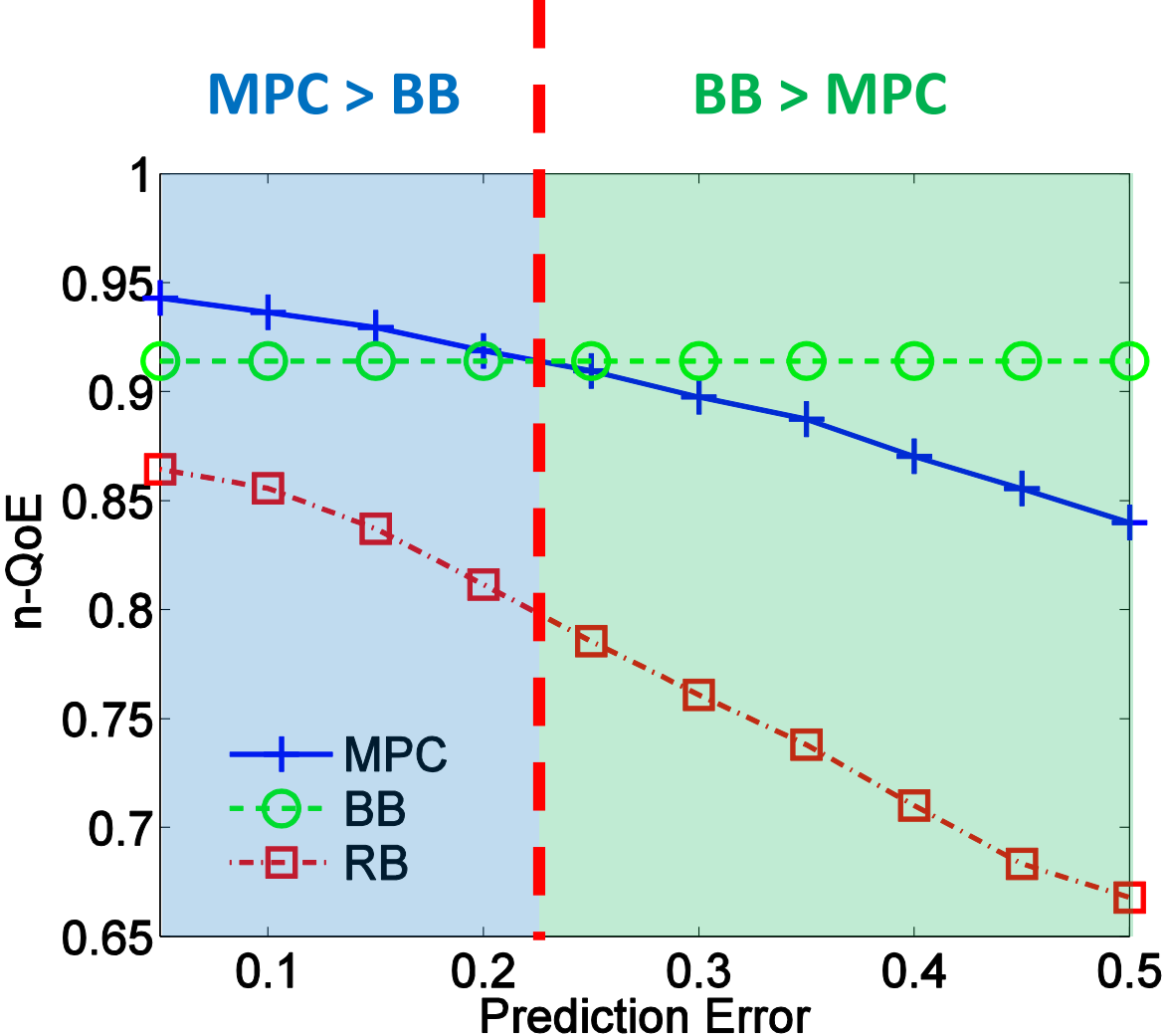


- Use both bandwidth and buffer information
- Smoothing out prediction error at each step
- Embed the control objective directly into controller

Main result: MPC > BB > RB



Does prediction error matter?



Summary of other insights

- All algorithms benefit from **finer-grained bitrate sets**
- MPC/BB can achieve **near-zero buffering** while RB cannot
- MPC do better on **avoiding bitrate variations**

Discussion & Limitations

- Full-spectrum sensitivity analysis
- Bandwidth estimation, interaction with TCP
- Characterizing bandwidth stability/predictability
- Multi-player interactions
- Computational complexity

Conclusions

- Lots of confusion in video player design
 - What is the objective? How to compare algorithms?
 - How sensitive is the solution?
- Use control theory to bring rigor to DASH design
- MPC outperforms BB and RB in certain conditions
- Future work: Bring control-theoretic framework to practice

Thank you!