A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP

Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, Bruno Sinopoli
Carnegie Mellon University
{yinxiaoqi522, abhishekjindal93}@gmail.com, {vsekar,brunos}@andrew.cmu.edu

Adaptive Video Player

- In the HTTP-based delivery model that predominates today, videos are typically chunked and encoded at different bitrate levels.
- The goal of an adaptive video player is to choose the bitrate level for future chunks to deliver the highest possible QoE
- Bitrate adaptation logic in the client-side video player becomes critical to optimize user experience

Video Streaming Model

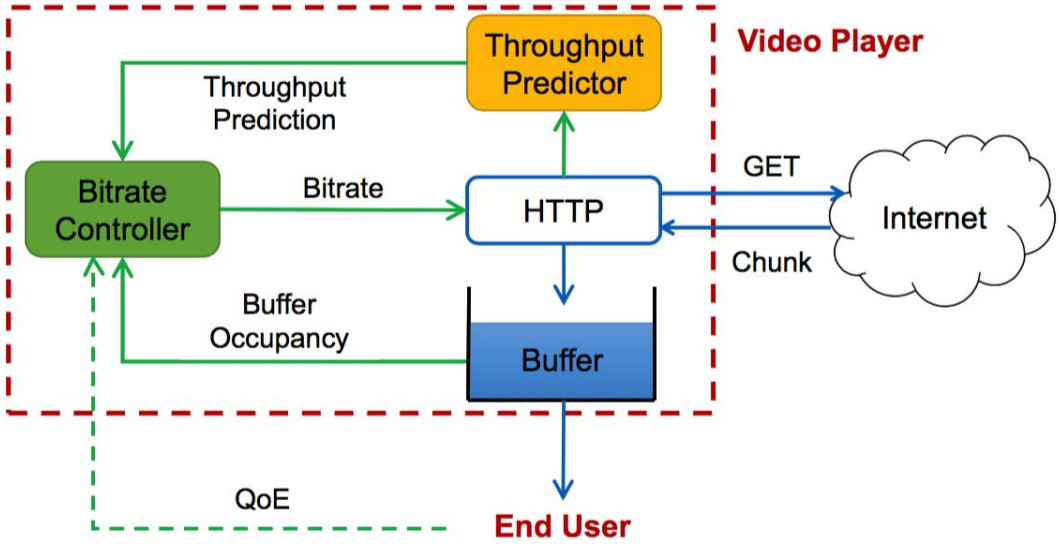


Figure 1: Abstract model of DASH players

Buffer Dynamics

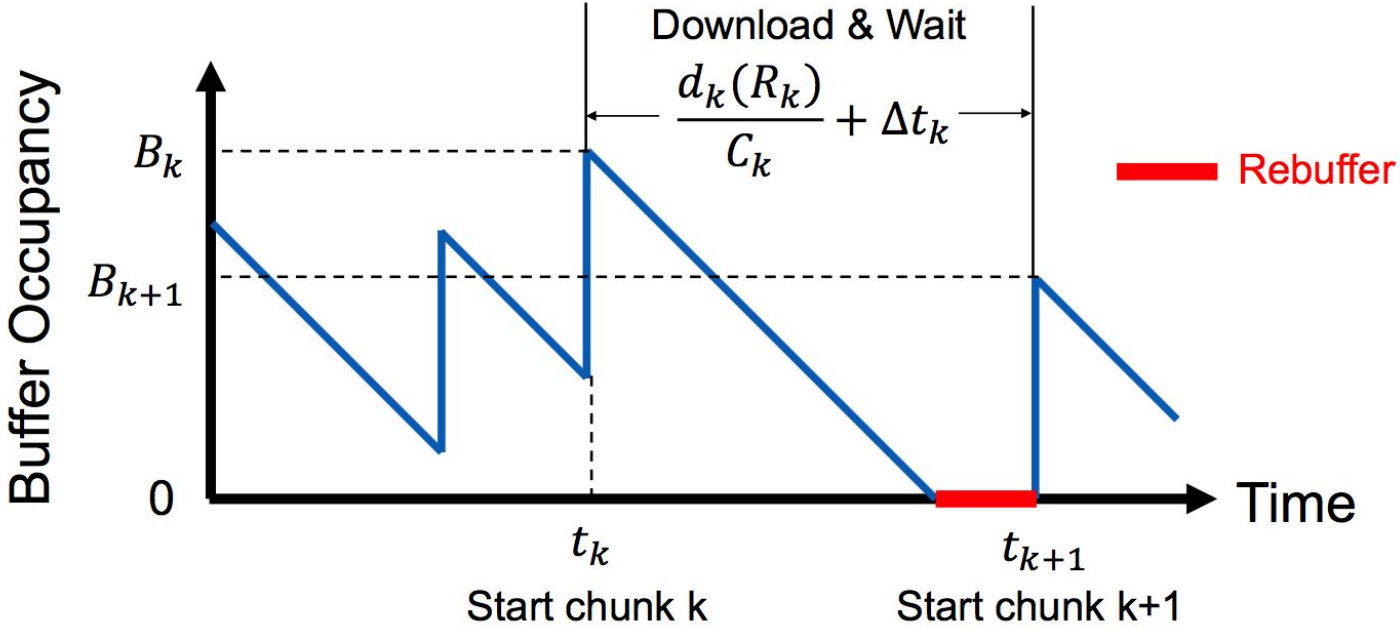


Figure 2: Illustration of buffer dynamics

Background: HTTP-based Adaptive Streaming

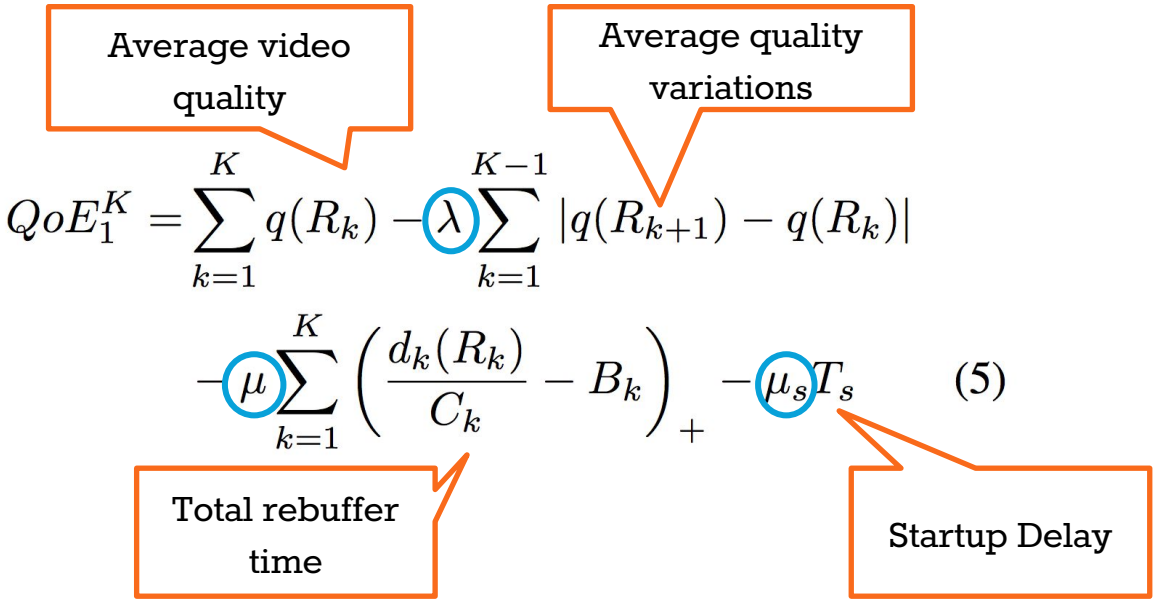
- Many conflicting goals:
  - minimizing rebuffering events where the playback buffer is empty and cannot render the video
  - delivering as high a playback bitrate as possible within the throughput constraints
  - minimizing startup delay so that the user does not quit while waiting for the video to load
  - keeping the playback as smooth as possible by avoiding frequent or large bitrate jumps

A client-side solution

- Why client-side:
  - Most immediately deployable alternative comparing to
    - network support
    - server-side support
    - lower-layer transport protocols
  - The client is often in the best position to quickly detect performance issues and respond to dynamics.

QoE Maximization Problem

- the QoE of video segment 1 through K by a weighted sum of the aforementioned components:



QoE Maximization Problem

$$\max_{R_1,\cdots,R_K,T_s} \quad QoE_1^K \tag{6}$$

$$s.t. \qquad t_{k+1} = t_k + \frac{d_k(R_k)}{C_k} + \Delta t_k, \tag{7}$$

$$C_k = \frac{1}{t_{k+1} - t_k - \Delta t_k} \int_{t_k}^{t_{k+1}-\Delta t_k} C_t \, dt, \tag{8}$$

$$B_{k+1} = \left(\left(B_k - \frac{d_k(R_k)}{C_k} \right)_+ + L - \Delta t_k \right)_+, \tag{9}$$

$$B_1 = T_s, \quad B_k \in [0, B_{max}] \tag{10}$$

$$R_k \in \mathcal{R}, \quad \forall k = 1, \cdots, K. \tag{11}$$

Figure 3: Formulation for QoE maximization ($QOE\_MAX_1^K$) subject to buffer and throughput dynamics.

QoE Bitrate Adaption

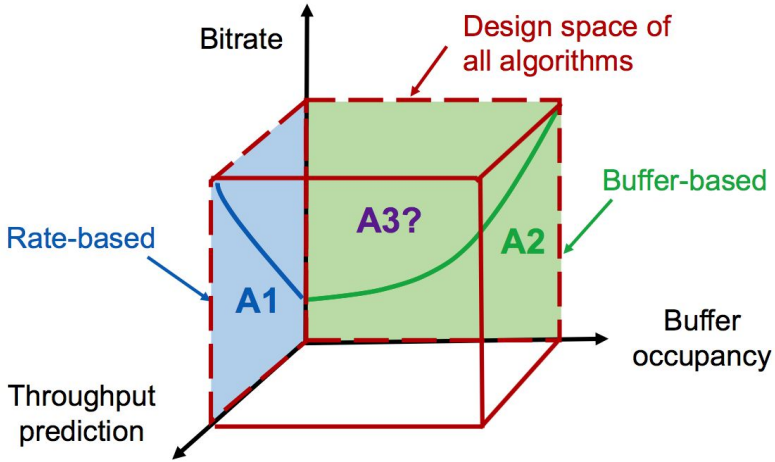


Figure 4: Design space of algorithms for the video adaptation problem: Most current approaches choose the bitrate as a function of only one variable; e.g., A1 is rate-based (RB) while A2 is buffer-based (BB).

Model Predictive Control

- Why MPC?
  - PID control is designed to work in continuous time and state space and using it in a highly discrete system such as ours may result in performance degradation or instability
  - with MPC (MDP) we could consider formulating the throughput and buffer state transition as Markov processes

Basic MPC Algorithm

Algorithm 1 Video adaptation workflow using MPC

Initialize
for $k = 1$ to K **do**
    if player is in startup phase **then**
        $\hat{C}_{[t_k, t_{k+N}]}$= ThroughputPred($C_{[t_1, t_k]}$)
        $[R_k, T_s] = f_{mpc}^{st}\left(R_{k-1}, B_k, \hat{C}_{[t_k, t_{k+N}]}\right)$
        Start playback after T_s seconds
    else if playback has started **then**
        $\hat{C}_{[t_k, t_{k+N}]}$= ThroughputPred($C_{[t_1, t_k]}$)
        $R_k = f_{mpc}\left(R_{k-1}, B_k, \hat{C}_{[t_k, t_{k+N}]}\right)$
    end if
    Download chunk k with bitrate R_k, wait till finished
end for

Robust MPC

- The basic MPC algorithm assumes the existence of an accurate throughput predictor.
- Robust MPC essentially optimizes the worst-case QoE assuming that the actual throughput can take any value in a range in contrast to a point estimate
- Robust MPC makes conservative estimation

FastMPC

- Why FastMPC?
  - MPC has large computational overhead
  - Video player cannot be bundled with the solver
- Solution: Table lookup

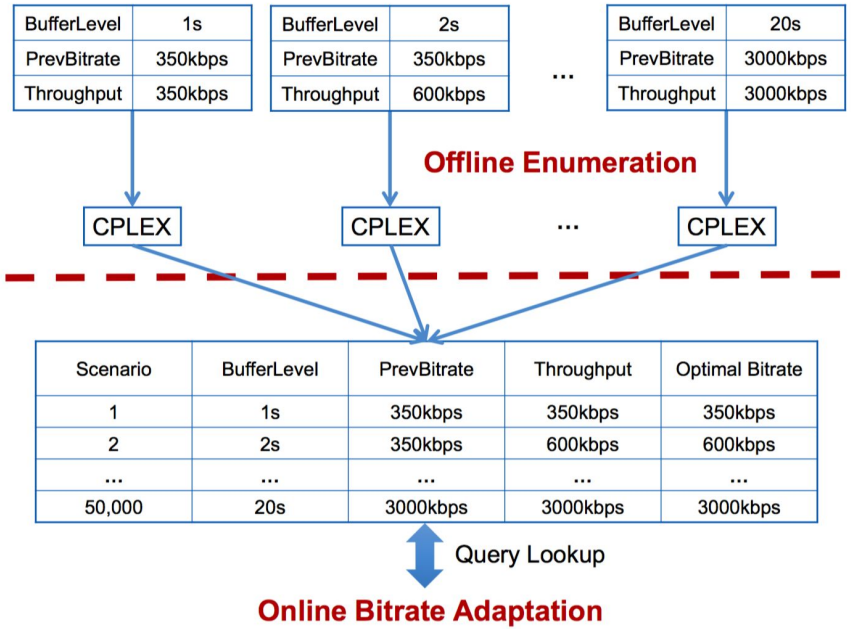
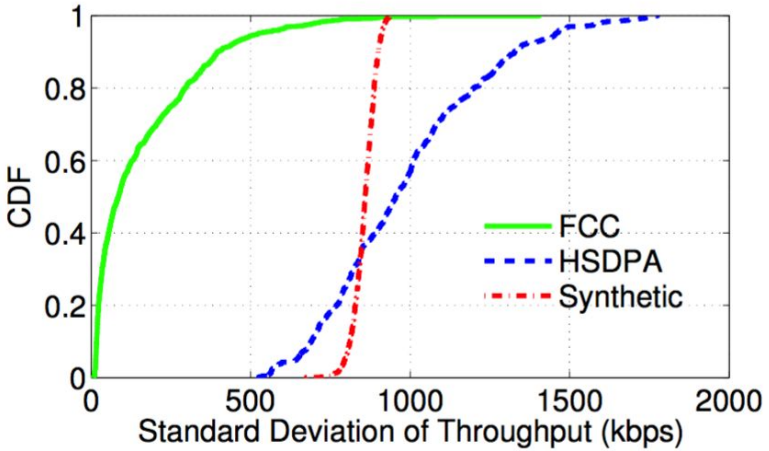
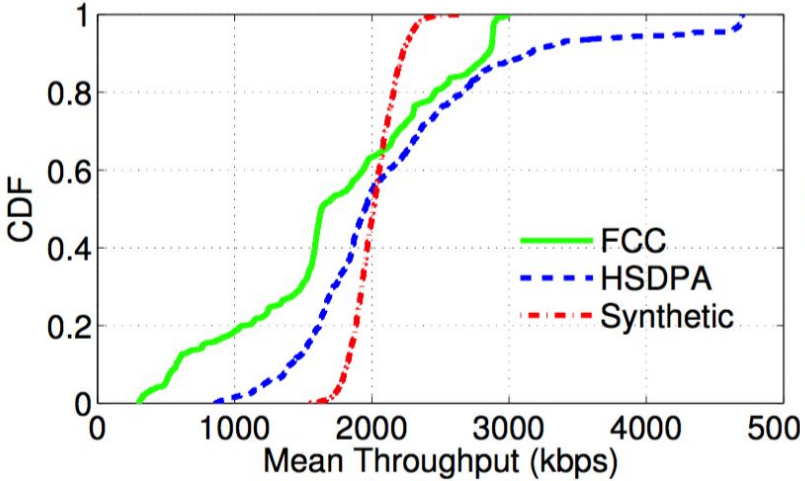


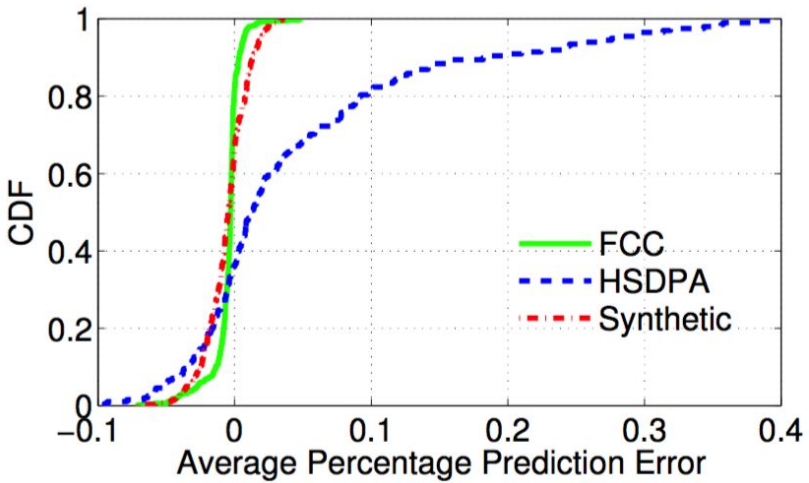
Figure 5: "FastMPC" idea: We enumerate possible scenarios and create a table indexing the optimal decision for each scenario.

Evaluation

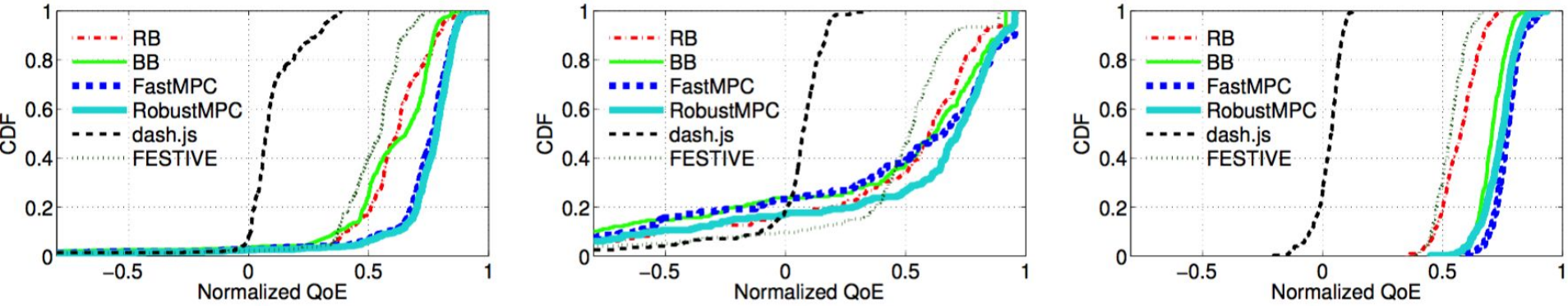
- FCC
- HSDPA
- Synthetic







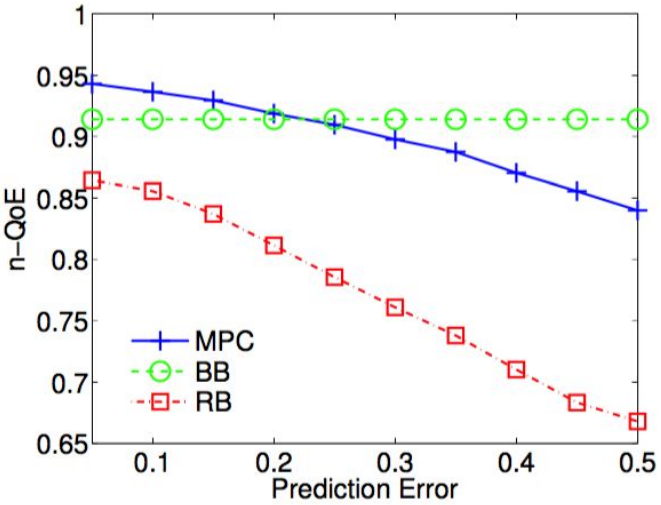
Real Player Evaluation



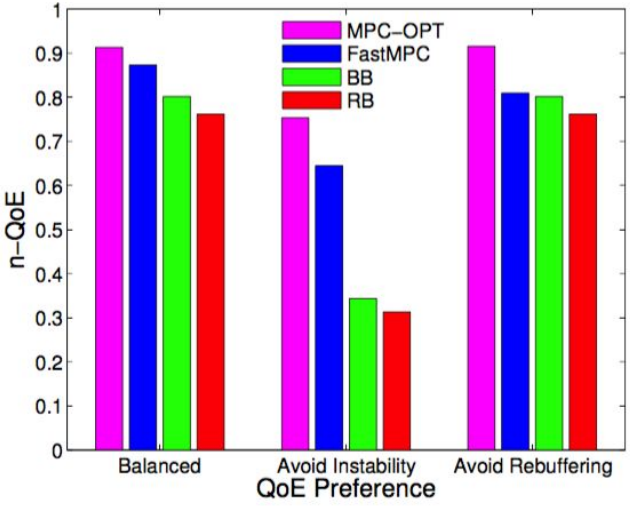
(a) FCC dataset (b) HSDPA dataset (c) Synthetic dataset

Figure 8: Real experiment results with different throughput traces

Sensitivity Analysis

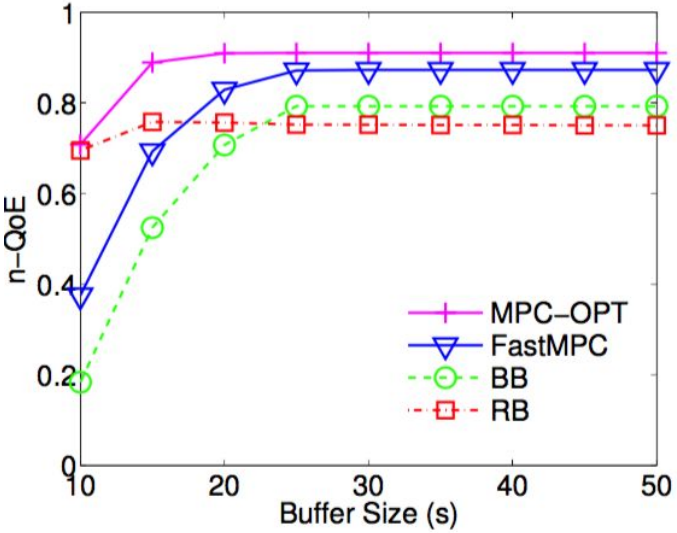


(a) Prediction error

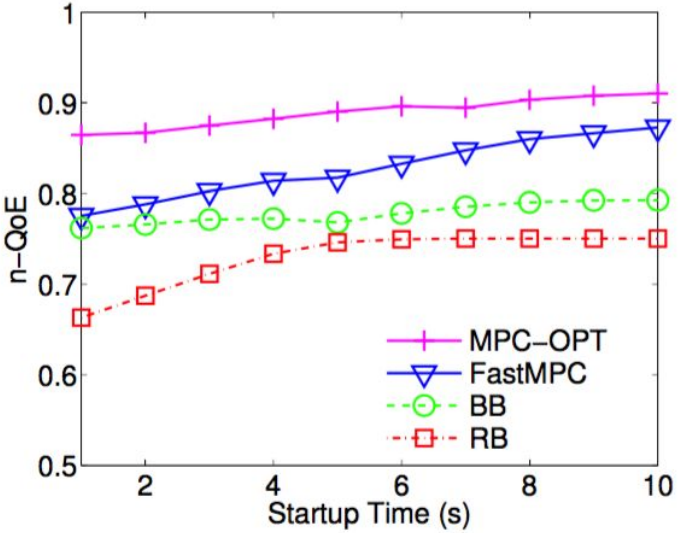


(b) QoE preferences

Sensitivity Analysis



(c) Buffer size



(d) Startup time

Takeaways and questions

Goal: developing a first-principles approach via control theory to develop a general framework to reason about classes of algorithms.

Questions to think about:

- What are some of the possible issues with MPC, comparing to BB and RB?
- What are the assumptions made by the paper that may not be realistic in practice?