# ECE 50863
# Project 3 Report Template

Name and email: Maximilian Drach (mdrach@purdue.edu)
Collaborator (if applicable): Timothy Hein (heint@purdue.edu)

# Base Algorithm: RobustMPC

| Feature | What you did? Focus on your implementation, and mainly how it might have simplified/modified what was in the paper. Detailed description of paper not needed. |
|---|---|
| What predictor did you use? | Lower Bounded Harmonic Mean using a dataset from the past 5 throughput values. |
| How did you adjust for prediction error? | Max Absolute Percentage Error $\text{err} = \max(\frac{\lvert Harmonic\ Mean - Measured\ Throughput \rvert}{Measured\ Throughput})$<br>To create the lower bound $\frac{Harmonic\ Mean}{1+err}$ |
| What look-ahead window did you use? | MDP look ahead was 5 chunks values |
| How do you solve the optimization for each look-ahead ? | I use a brute force method which covers all possible product combinations.<br>$(Number\ of\ Bitrate\ Levels)^{Number\ of\ Look-Ahead\ Chunks}$ |
| Do you recompute bitrate choices at each chunk or every few chunks? [E.g., chunks 1-5, 2-6, 3-7 etc. or 1-5,6-10,11-15] | At every chunk we recompute the bitrate level choice. |

# BaseAlgorithm: RobustMPC

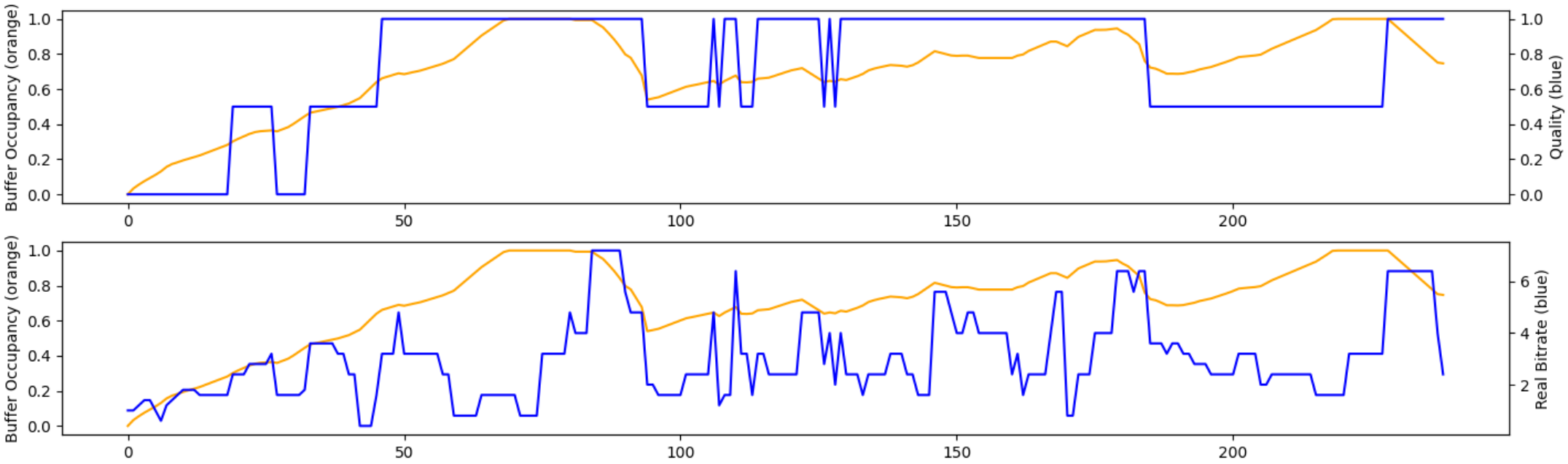
- In our RobustMPC implementation we do not optimize the start-up time parameter. Since we have no previous throughput data, it is hard for us to optimize an ideal start-up time with out more data. Therefore, in our implementation I pick the lowest bitrate level for the initial chunk request.
- We optimize for average video quality and average quality variation using the absolute bitrate level. While we use the chuck bitrate to optimize the rebuffer factor.
- For optimizing the rebuffer factor, we also make sure the value is a minimum of 0 thus not differentiating value that do not decrease the buffer level.
  - For our improve approach we change this minimum value to -10.
- We simplify the RobustMPC by not accounting for the waiting time in between chunks request.

# Base Algorithm: BBA-2

| Feature | What you did? Focus on your implementation, and mainly how it might have simplified/modified what was in the paper. Detailed description of paper not needed. |
|---|---|
| How was initial period handled to avoid being conservative | We used a special “startup period” in which the quality level is based on a how quickly the buffer is replenishing. While the buffer is within the reservoir it ups the quality if a chunk is downloaded within ⅛ the time it takes to play, above the reservoir ½. The startup period ends if the upper reservoir is reached or if the buffer decreases between chunks |
| How did you map buffer levels to chunk sizes? | We measure the buffer in time remaining before a rebuffer occurs. We then modulate the reservoir based on having a certain number of chunks in the buffer instead of a certain number of bytes. |

# BaseAlgorithm: BBA-2

- We implement a smoothing function that waits until the variable bit rate is sufficiently below the nominal for next several chunks before allowing an increase in quality. This ensures the algorithm only changes when it expects it will be able to sustain a full buffer.



# Your approach [Use more slides if needed]

For checkpoint, this is a proposed approach. For final replace with what you actually did.

- The MPC paper implementation does not differentiate the positive impact a bitrate level has on the buffer level. This mean that any bitrate that doesn’t negatively affect the buffer level is treated the same in the MPC equation (ie the rebuffering term has a minimum of 0.) In our implementation we allow a bitrate level to positively impact the buffer level up to 10 seconds in the MPC equation (ie the minimum is -10 seconds.) We hope this reduces the number of rebufferings by enabling the MPC to expands the scope it allows bitrate level buffer level impacts.

# MPC QoE Scores

## Positive Buffer Factor Up to 0 secs (Average QoE = 1.49)

| Throughput | Variation | Total Quality | Total Variation | Rebuffer Time | Total QoE |
|---|---|---|---|---|---|
| High | High | 405 | 53 | 1.13 | 3.142689 |
| High | Medium | 430 | 30 | 5.12 | 3.315294 |
| High | Low | 394 | 28 | 0.75 | 3.168067 |
| Medium | High | 281 | 76 | 12.67 | 1.616134 |
| Medium | Medium | 237 | 50 | 27.33 | 0.862857 |
| Medium | Low | 276 | 77 | 9.2 | 1.686555 |
| Low | High | 189 | 76 | 37.14 | 0.020504 |
| Low | Medium | 125 | 76 | 25.46 | -0.12471 |
| Low | Low | 147 | 80 | 33.71 | -0.23395 |

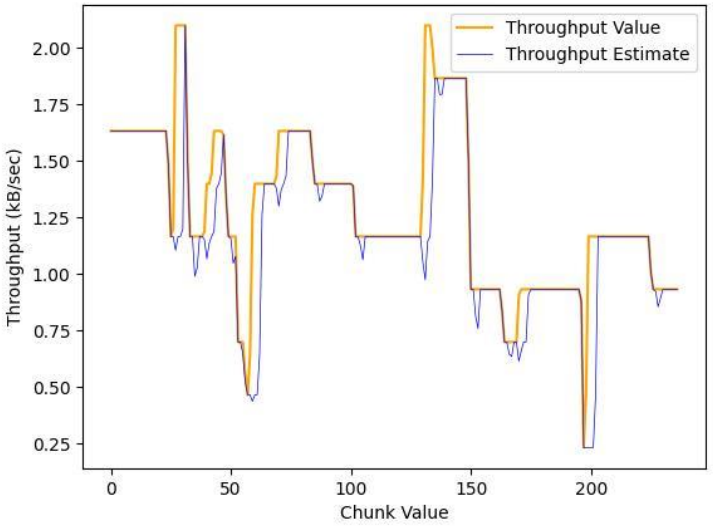
## Positive Buffer Impact Up to 10 secs (Average QoE = 1.71)

| Throughput | Variation | Total Quality | Total Variation | Rebuffer Time | Total QoE |
|---|---|---|---|---|---|
| High | High | 385 | 38 | 0.33 | 3.064538 |
| High | Medium | 398 | 34 | 0.33 | 3.190588 |
| High | Low | 380 | 30 | 0.47 | 3.051429 |
| Medium | High | 248 | 44 | 0.4 | 1.885714 |
| Medium | Medium | 182 | 35 | 3.93 | 1.250252 |
| Medium | Low | 232 | 68 | 0.5 | 1.647059 |
| Low | High | 151 | 78 | 1.12 | 0.903529 |
| Low | Medium | 56 | 64 | 0.95 | 0.169748 |
| Low | Low | 101 | 40 | 13.43 | 0.229244 |

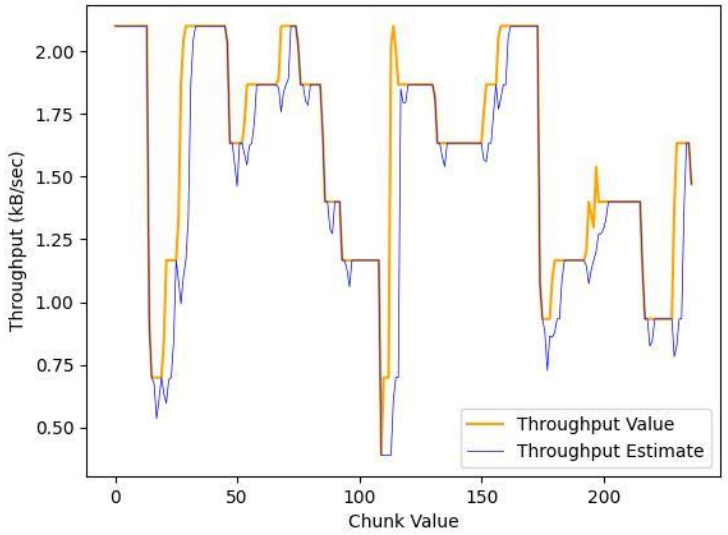
- We had a hypothesis that MPC was unable to perform as good in Low Throughput environments because it could not gauge the positive impact of lower bitrate levels to refill the buffer and thus leading to high rebuffer times. To test our hypothesis, we switched the minimum value a rebuffering factor could be from 0 to -10, this would allow the MPC to account for bitrate levels that can positively increase a buffer size by 10 seconds. Incorporating the positive impact of bitrate levels can positively impact the buffer size by enabling a faster download of specific chunks and thus growing the buffer level in the long term.
- As we can see from the tables the lower rebuffering times are clearly lower and thus greatly improving the Lower Throughput QoE. While we see the Total Quality score decrease, it only affects the QoE score of the High Throughput environments where we see a slight drop.
- Overall, this minor change clearly has net positive impact on the end QoE due to its impact on reducing rebuffering time.

# Low Throughput Estimate vs Measure Value

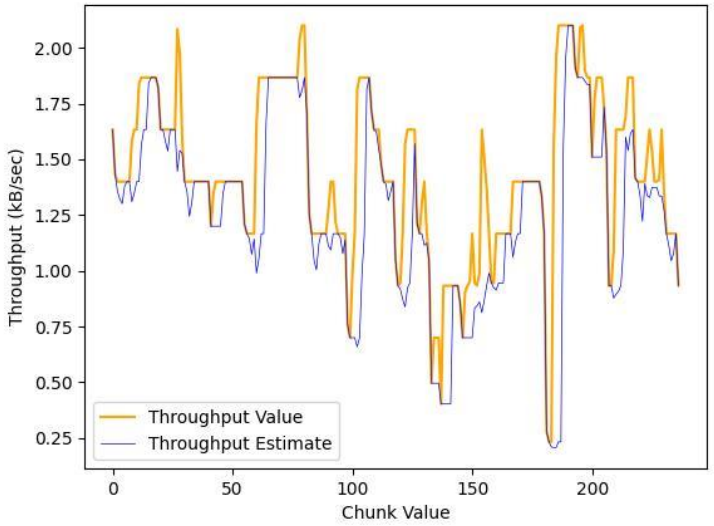
Low Variation




Medium Variation




High Variation




- As we can see from the plots above the harmonic mean does a pretty good job at estimating the throughput.
- While future implementation might include a better throughput predictor, the stochastic nature of throughput activity means we will probably focus our activities on future optimizing the bitrate process not predicting the throughput better.

# Contributions of each student

[Skip slide if you did this alone;  for checkpoint indicate ”Completed” and “Proposed” for each bullet.]

- Max – Robust MPC
- Timmy – BBA2