Questions:

1) **What are the key factors that constitute the Quality of Experience (QoE) of a video session?**

   Duration of rebuffering, Startup delay, Average playback bitrate, and Variability of the bitrate delivered.

2) **As per the paper, what are the two categories of existing solutions for video streaming? Why do they fall short?**

   Rate-based algorithms (cons: throughput estimation on top of HTTP suffers from significant biases) and Buffer-rated algorithms (cons: discard available throughput information). They fall short since both of them offers point heuristics that work under specific (and implicit) environmental assumptions.

3) **At each step, the algorithm solves an optimization problem shown in Figure 3. What are the variables that it is trying to determine? What is the objective?**

   The algorithm is trying to determine "Bitrate decisions" and "Startup time". The objective is to maximize the QoE of the weighted sum of several components to model varying user preferences on different contributing factors.

4) **The optimization in Figure 3 needs C_t, a prediction of future throughput at time t.**
   What prediction method did the authors use in this paper? [Hint: end of Sec 6, and 7.1.2]

   Harmonic mean based throughput prediction scheme of the observed throughput of the last 5 chunks is used in this paper.

5) **What are the main reasons the authors chose a Model Predictive Control approach?**

   PID control can only serve to stabilize the system and cannot explicitly optimize the QoE objective. Besides, MDP-based control needs an unpractical assumption that throughput dynamics follow Markov processes. Hence, with the possibility that reasonably accurate throughput prediction can be obtained for a short horizon, MPC is chosen.

6) **What is RobustMPC? What is the disadvantage of RobustMPC? What is the advantage of RobustMPC?**

   RobustMPC optimizes the worst-case QoE assuming the actual throughput can take any value in a range (in contrast to a point estimate). The disadvantage is that it's more conservative than regular MPC. The advantage is that it could better handle errors in throughput prediction under highly variable network conditions.

7) **Based on the evaluation section (7.1.2), what is the main difference between the FastMPC algorithm and the RobustMPC algorithm? [Note: don't worry about the techniques for computation efficiency for FastMPC described earlier in the paper for this question].**

   Both use the same throughput prediction scheme (harmonic mean of past 5 chunks). But FastMPC uses table enumeration with a look-ahead horizon (h=5) to determine the throughput. On the other hand, RobustMPC assumes the throughput lower bound via the throughput prediction and MAX absolute percentage prediction error.

8) **In the evaluations, what is the QoE metric normalized to for each algorithm?**

The QoE metric is normalized for each algorithm by the maximum QoE that can be achieved with perfect knowledge of future throughput over the entire horizon.

9) **In the evaluations, what tools are used to emulate different network conditions?**

Linux 'tc' tool is used to throttle the throughput of the link between two computers according to the throughput traces employed.

10) **The authors use three sets of throughput traces. What are the key differences in characteristics of these three traces, that may impact a relative comparison of different algorithms?**

The key differences in characteristics of the three throughput traces are throughput variability under network variability conditions and average percentage prediction error. For example, throughput is the most stable in the broadband network (FCC) and the most variable in mobile networks (HSDPA).

11) **How the FastMPC and RobustMPC algorithms compare across the three traces for the Normalized QoE metric? Is one always better? If so, why? If not, which performs better for which trace, and why?**

There is no difference between FastMPC and RobustMPC on FCC and Synthetic traces. However, in the HSDPA trace, RobustMPC performs better than FastMPC. This is because FastMPC is more sensitive to prediction errors than RobustMPC, and HSDPA has higher average percentage prediction error compared with FCC and Synthetic.

12) **In Figure 8, the authors present the QoE metric, while in Figure 10, the authors present results with many individual metrics (average bitrate, rebuffer time, and bitrate change). How does the RobustMPC algorithm perform relative to other algorithms in these individual metrics? How does it perform relative to other algorithms on the normalized QoE metric?**

In the FCC dataset, RobustMPC, FastMPC, and BB share similar average bitrate, but RobustMPC has fewer bitrate changes. In the HSPDA dataset, RobustMPC achieves significantly less rebuffer time but at a slightly lower average bitrate. As a result, RobustMPC outperforms other algorithms in overall normalized QoE.

13) **Open-ended: Based on the paper, do you have any thoughts on a better algorithm that you could design? Or any thoughts on a potential problem that the proposed MPC approach may encounter? [Not more than 4-5 lines].**

This paper cites a paper in 2001 to indicate network conditions are reasonably stable on short timescales. But the timescale is tens of seconds. I'm wondering if this statement is still valid nowadays. Besides, I think the table enumeration approach of FastMPC might could be adjusted and improved to cover more realistic scenarios.