Lab of Computer Network: Video Streaming **Fall 2024**

Report 15 — December 14

Lecturer: Wu Qinghua *Completed by: 2022K8009929010 Zhang Jiawei*

15.1 实验内容

- 安装 selenium，用于自动化执行视频请求和播放，遇到询问是否使用 snap 安装浏览器时，直接 skip
  - `sudo apt install python3-numpy python3-selenium`
- 安装浏览器和驱动：如果 Ubuntu 已经使用 snap 安装了 firefox 浏览器，先卸载，再从 mozilla 源安装：
  - `sudo snap remove firefox`
  - `sudo apt remove firefox`
  - `./setup-mozilla-repo.sh`
  - `sudo apt update && sudo apt install firefox`
- 从https://github.com/mozilla/geckodriver/releases下载并安装驱动 geckodriver
  - `tar xzvf geckodriver-v0.xx.y-linux64.tar.gz` # 下载最新版，macbook 选择 aarch64 版本
  - `sudo cp .../geckodriver /usr/bin/` # ... 代表解压后的文件夹
- 将 15-video-streaming.tar.gz 文件进行解压
  - `tar xzvf 15-video-streaming.tar.gz`
- 执行 run_abr_exp.py 脚本，指定 ABR 算法
  - `sudo python3 run_abr_exp.py BB` # alternatives: RB, MPC
- 上述脚本大概运行 120 秒，获取并播放 100 秒视频，将过程中的码率、卡顿等日志信息写入 results 文件夹
- 每行日志对应该 ABR 算法获取一个视频块时的信息，格式如下：
  - 时间戳，码率（Kbps），buffer（s），卡顿时间（s），块大小（B），下载时间（ms），QoE，带宽（Kbps）
- 逐个运行以下 ABR 算法：BB, RB, MPC

15.2 实验结果

逐个运行之后，画图如下：

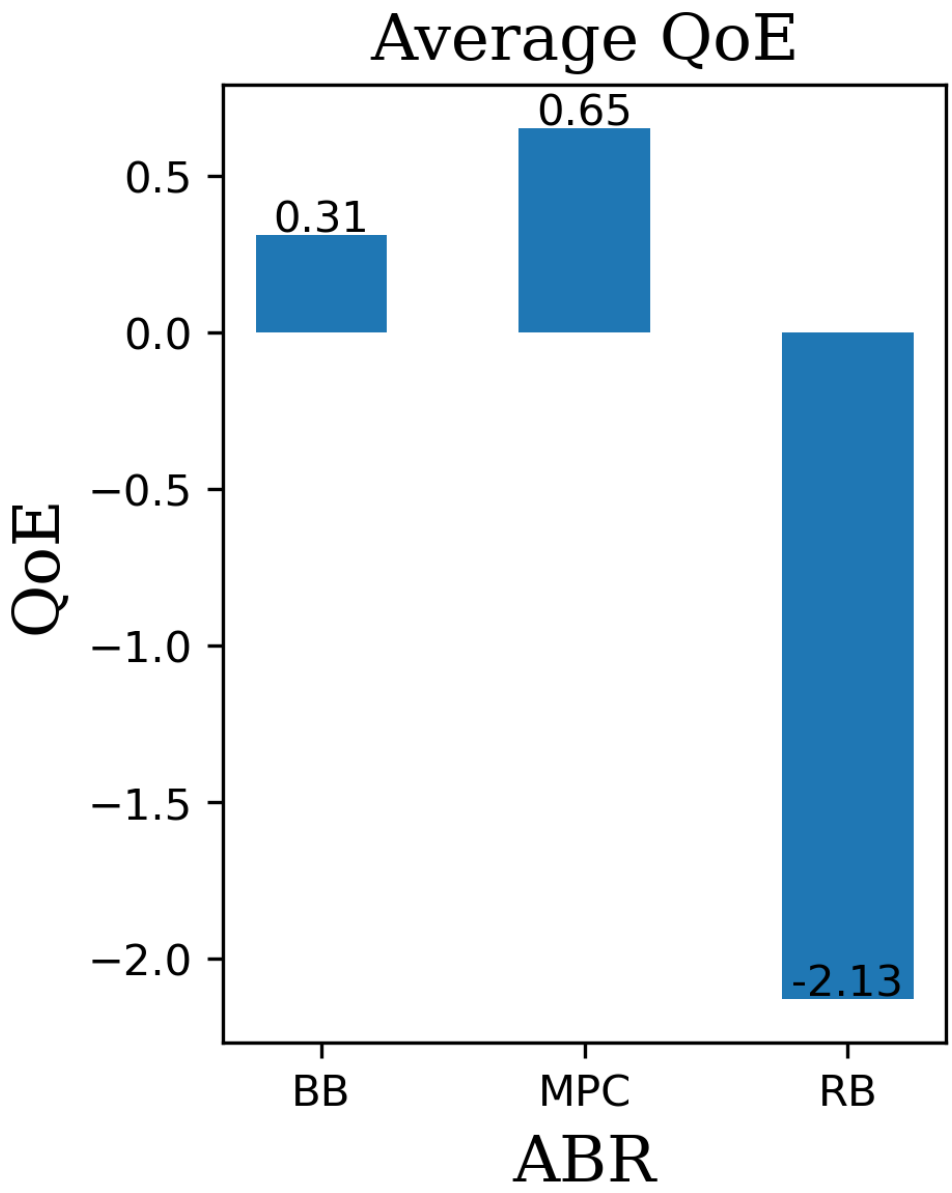

图 **15.1.** QoE 对比

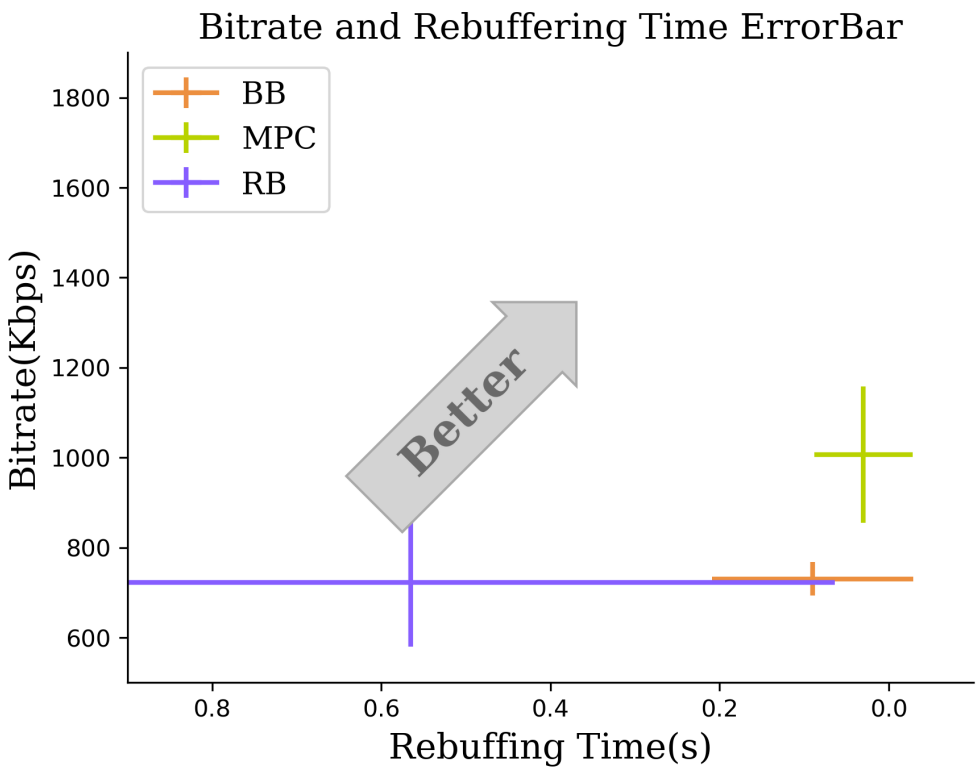

图 **15.2.** Buff 重填时间与比特率关系图及对比

从图中可以看出，MPC 算法的 QoE 最高，其次是 BB 算法，最后是 RB 算法。而在 Buff 重填时间与比特率关系图中，MPC 算法的重填时间最短且比特率最高，BB 算法次之，RB 算法最差。

产生这种结果的原因大概是，MPC 算法是基于最优控制理论的，可以根据当前网络状态和视频播放状态来选择最优的码率，以达到最佳的 QoE。而 BB 算法是基于带宽预测的，根据预测的带宽来选择码率，因此 QoE 次之。RB 算法是基于带宽预测和缓冲区状态的，但是由于缓冲区状态的更新不及时，导致 QoE 最差。

15.3 实验总结

本次实验主要是通过自动化脚本运行 ABR 算法，获取视频播放过程中的日志信息，然后通过画图对比不同 ABR 算法的 QoE 和 Buff 重填时间与比特率关系。从结果来看，MPC 算法的 QoE 最高，且重填时间最短，比特率最高，说明 MPC 算法在视频流媒体中表现最好。我也通过这次实验学习到了如何使用 selenium 自动化执行浏览器操作，以及如何获取视频播放过程中的日志信息，对于视频流媒体的 ABR 算法有了更深入的了解。