

## Supporting Time-Sensitive Applications on a Commodity OS

This paper seeks to create an operating system that can satisfy the demands of time-sensitive applications. It notes that most typical operating systems do not meet these demands due to having higher timer latency, kernel latency, and scheduling latency. There are three main modifications that can be made to a base commodity OS, like Linux, that the paper claims will allow it have more fine-grained allocation of resources that time-sensitive applications require while also still being efficient for throughput-oriented tasks. These modifications are using firm timers to reduce timer latency, making a preemptible kernel, and implementing real-time oriented CPU scheduling algorithms such as proportion-period. The paper then goes through the technical design and implementation of these three features and then moves to evaluating the new OS's performance. They perform a number of benchmarks testing how well an application that synchronizes video and audio can perform under various CPU conditions. Their tests show that in every case, their time sensitive Linux allows the application to more consistently sync audio and video without delay, in comparison to normal Linux.

One of the claims of the paper (that's even in the title) is that their time-sensitive Linux would be able to meet the requirements for time-sensitive applications while also still being able to manage throughput-oriented applications, so that the OS could be used not jut in niche situations, but rather in a general-purpose way. However, the paper doesn't really support this claim. From everything they described in the technical implementation section, there's not much reason to believe their OS can't have good performance on throughput-oriented tasks, but there's no empirical evidence for it in the same way there's many benchmarks and graphs showcasing the performance of their OS on a time-sensitive application. One improvement I would suggest is a small section that just compares the time to complete a throughput-oriented task on base Linux and time-sensitive Linux, just to have some assurance they get similar performance.

A notable strength of the paper is their explanation of all the different changes needed to accommodate time-sensitive applications and why those changes would help. The audience for this paper is probably supposed to be someone with more knowledge about operating systems than me, but even with my limited knowledge I was able to follow the reasoning well. Another strength is their experimental design for testing on performance of time-sensitive Linux. The idea of using three different background loads to isolate various changes they made, such as non-kernel CPU load to test timer latency and kernel CPU load to test kernel preemptibility, was clever.