RT USB Streaming Protocol

Ross Mikulskis

February 4, 2025

Contents

1 Introduction

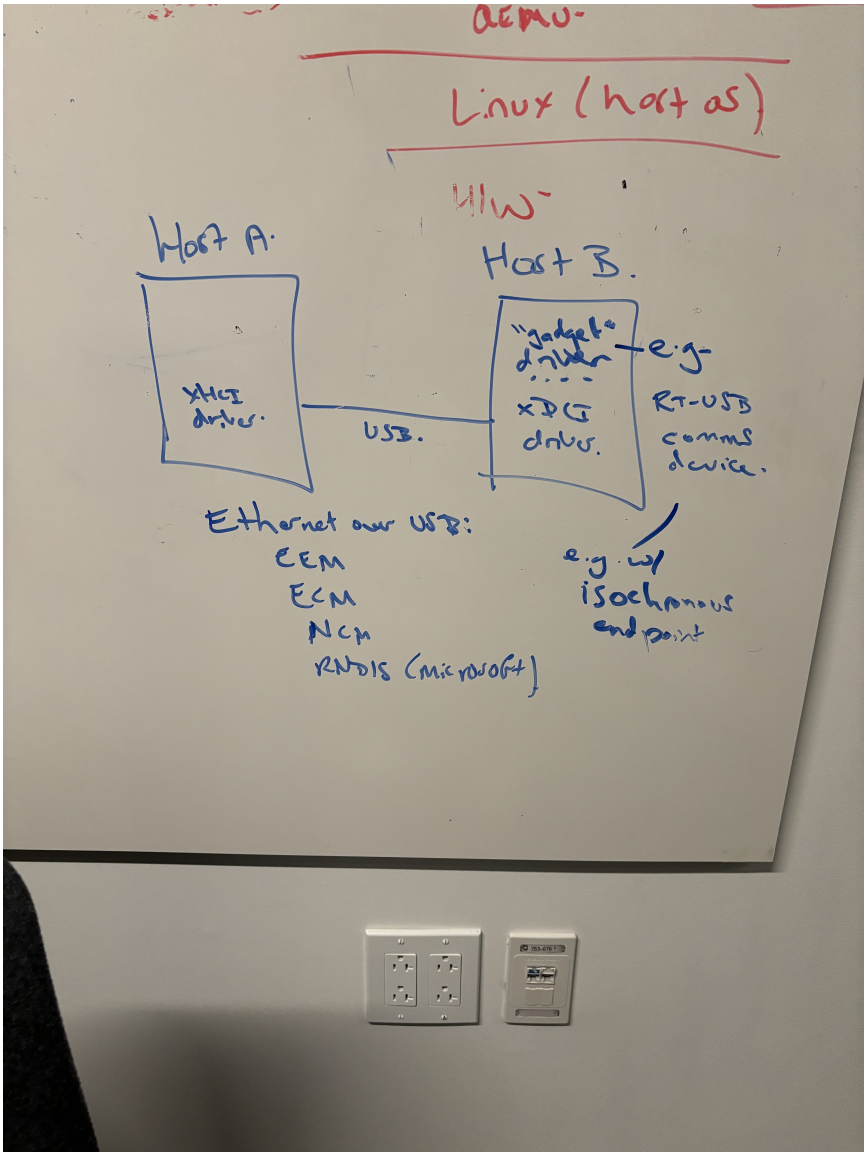
Quest is a Real-Time Operating System (RTOS) that supports scheduling tasks in an isolated, predictable manner. We've developed RT-DDS, a publisher-subscriber framework which connects services over shared memory communication channels in pipelines to ensure end-to-end latency, throughput, and loss requirements. Each service in RT-DDS is a tuned-pipe with an input and output endpoint and thread bound to a VCPU (Virtual CPU) with a corresponding budget C and period T using rate-monotonic scheduling (RMS).

Similarly, we can schedule USB bandwidth reservations with a budget and period over isochronous endpoints for real-time USB communication. Currently Quest only supports bulk transfer endpoints, and these endpoints cannot be scheduled for real-time transactions. Therefore, bulk transfers may be stalled on account of pending isochronous requests on the host, which may be on other USB lines.

Using xDCI (Extensible Device Controller Interface) drivers, we can emulate a device over USB using a gadget driver, for example Linux supports Ethernet over USB from host to host.

2 Project goal

We want to finish the implementation of the xDCI driver on Quest to support isochronous endpoint communication. Using the xDCI driver, we plan to write a real-time USB gadget device driver with our own custom protocol. We can then extend RT-DDS for cross-host pipelines. See the image below which depicts two hosts communicating over an isocrhonous USB endpoint.



3 Tasks to complete

1. finish implementation of the xDCI driver on Quest and
2. develop a RT-USB gadget device driver to support isochronous endpoints on Quest

3. (reach goal) support RT-DDS pipelines spanning hosts over USB using this RT-USB gadget device driver
    - Incorporate USB latency calculations in RT-DDS scheduler

4 Required skill set

- Knowledge of xDCI USB spec
- Embedded systems programming skills, C coding

5 Resources

USB Interrupt Differentiated Service for Bandwidth and Delay-Constrained Input/Output