Advanced Operating Systems:
Lab 3 – TCP
ACS/Part III Assignment

Prof. Robert N. M. Watson

2023-2024

This is Part III/ACS Lab 3. If you are an Part II student, please see the other lab variant.

Your lab report will analyse how network latency impacts TCP throughput, with a particular interest in its effects on congestion control.

1 Approach

You will run a series of experiments using the IPC benchmark using the `tcp` IPC type, using DUMMYNET to simulate varying network latency. This lab requires you to consider only the TCP steady state (`ESTABLISHED`), and not the three-way handshake or connection close. Throughout, use a fixed buffer size of 1MiB (`-b 1048576`).

Impose simulated latencies through the use of two DUMMYNET pipes, one for each traffic directly, dividing latency evenly over the two pipes. That is, for a 10ms simulated latency, introduce 5ms of latency on each pipe. Use DTrace and other tooling to explain the effects you see, with respect to the assignment hypothesis.

Submitting your completed assignment

Your submitted lab report will be a single PDF file using the the ACS/Part III lab-report LaTeX provided. All submissions are via the course's Moodle page.

2 Baseline bandwidth measurements

Measure and plot the bandwidth reported by the benchmark at three simulated round-trip latencies of 0ms, 10ms, 20ms. Use five iterations for each configuration to allow you to measure variance. The plot will have an X axis for latency, and Y axis for achieved bandwidth. Describe how varying latency has impacted performance.

3 Analyzing specific runs

Use DTrace to instrument the execution of three individual benchmark runs at 0ms, 10ms, and 20ms latency, using one iteration each. Collect and plot data that allows you to explain how varying latency interacts with congestion control to cause this performance variation:

1. Plot your results using a TCP time-bandwidth plot, which places time on the linear X axis, and bandwidth achieved by TCP on a linear or log Y axis. Bandwidth may be usefully calculated as the change in sequence number (i.e., bytes) over a window of time – e.g., a period of 1 second. Care should be taken to handle wrapping in the 32-bit sequence space; for shorter measurements this might be accomplished by dropping traces from experimental runs in which sequence numbers wrap.

2. Overlay additional time-based data, such as specific annotations of trace events from the congestion-control implementation, such as packet-loss detection or transition into and out of slow start. Rather than directly overlaying, which can be visually confusing due to conflating Y scales, a better option may be to "stack" the graphs: Place them on the same X axis (time), horizontally aligned but vertically stacked. Possible additional data points (and Y axes) might include advertised and congestion-window sizes in bytes.

Using this data, explore and validate our first hypothesis, explaining your conclusions.

4 Analyzing the probe effect

Unlike our IPC workload, TCP's behavior is sensitive to specific execution timing. This raises a greater risk that the instrumented workload does not behave in the same way as the uninstrumented workload. Experimentally explore the impact of the probe effect by comparing instrumented and uninstrumented runs of the benchmark at 0ms and 20ms latencies. Plot the results along with other metrics of interest, and explore the hypothesis that, despite the probe effect, your argument above relating to latency impact on performance is valid.