Debugging Heterogeneous Hardware Research Experiments on a Large-Scale Network Testbed

ALEXANDER WOLOSEWICZ (IIT), VINOD YEGNESWARAN (SRI), ASHISH GEHANI (SRI),

KOMAL THAREJA (RENCI), MERT CEVIK (RENCI), NIK SULTANA (IIT)

Designing the Debugger

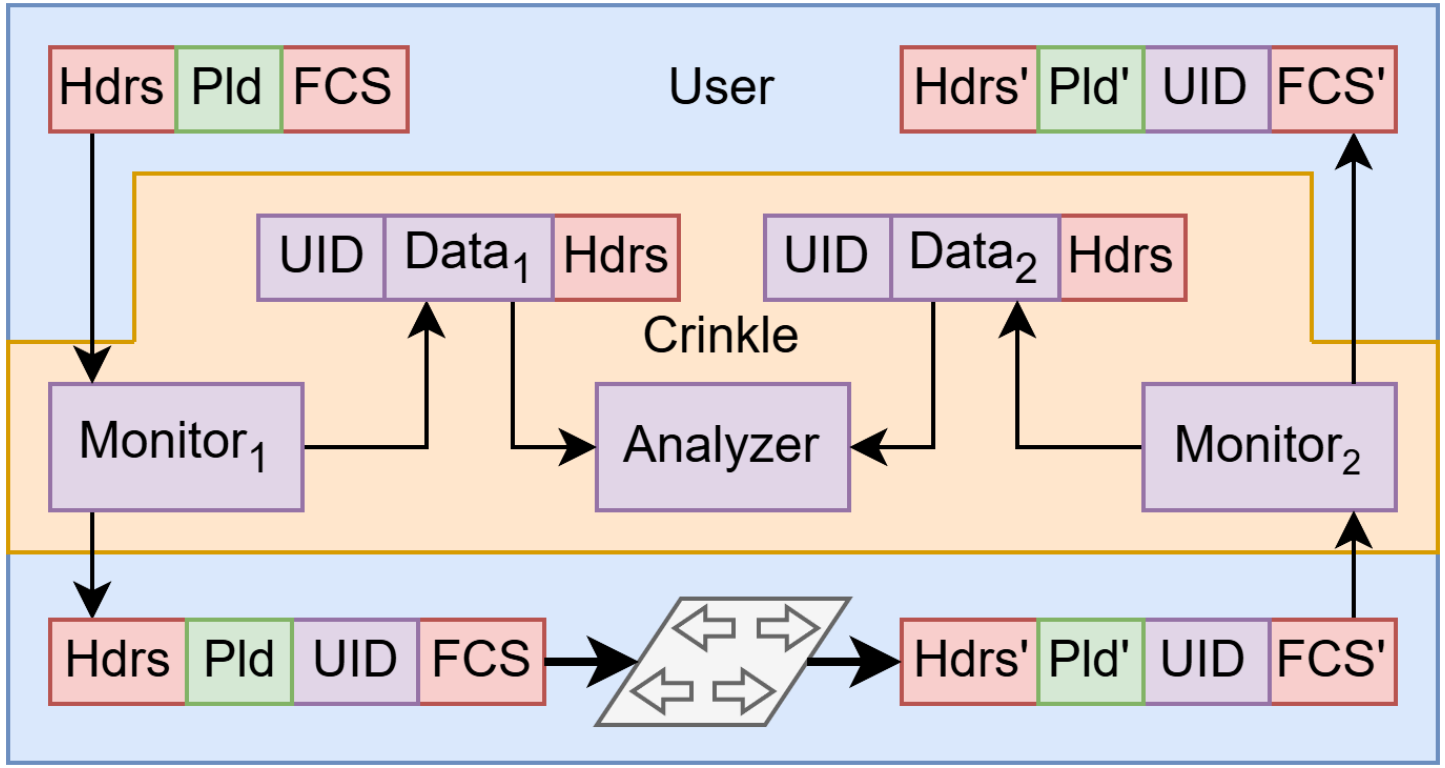
FABRIC is a testbed – freely insert VMs

Use these VMs to monitor and manipulate traffic, but abstract them from the experimental topology

- To the user, topology appears to function largely as if the debugger isn't there

These monitors insert a trailer that identifies the packet, enabling tracing the history of a packet across the network

Designing the Debugger

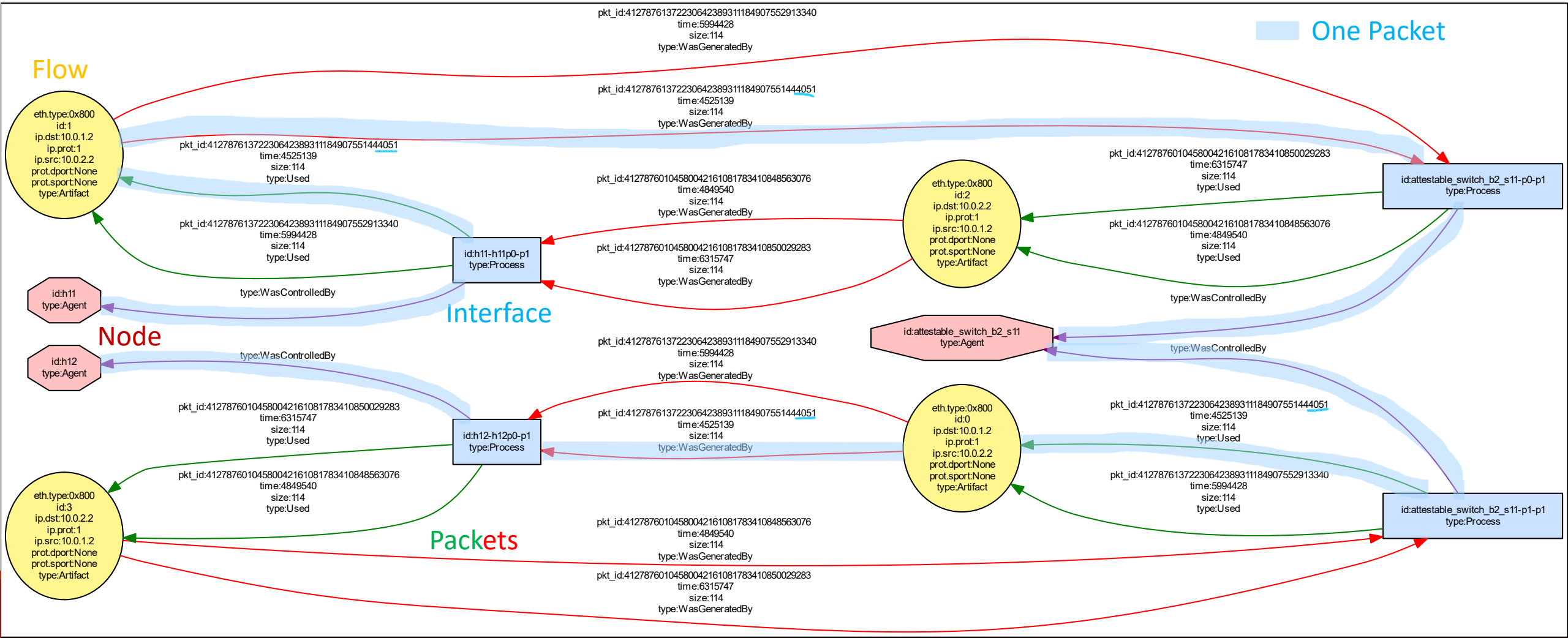


Designing the Debugger

Collected packet histories stored in SPADE

Query-able, and produces graph visualizations

Designing the Debugger



Using the Debugger

Monitors inserted into the experimental topology

Can query the history database by time, packet id, flow attributes, location

Can craft probe packets and retrieve their specific history

Foundation for future features that approach what is capable in software debuggers

(Preliminary) Statistics

Carried out on the FABRIC STAR site

Manually allocated onto less-utilized workers

Two flows of 200 packets, 100 each direction per flow

DPDK 24.11 running in Ubuntu 22.04

Statistics

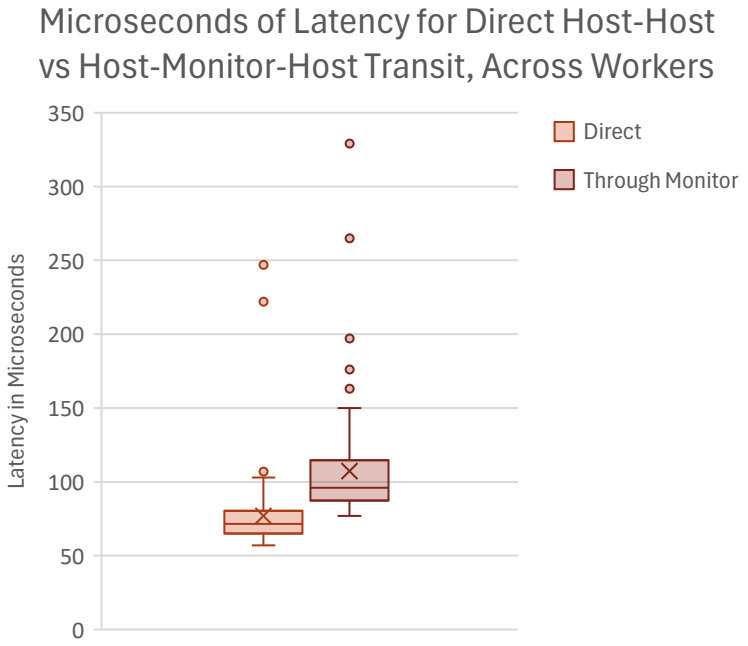
Delta Min/Avg/Max: +20/30/82 μs

Avg processing overhead: 2.75 μs

Statistics

Delta Min/Avg/Max: +20/30/82 μs

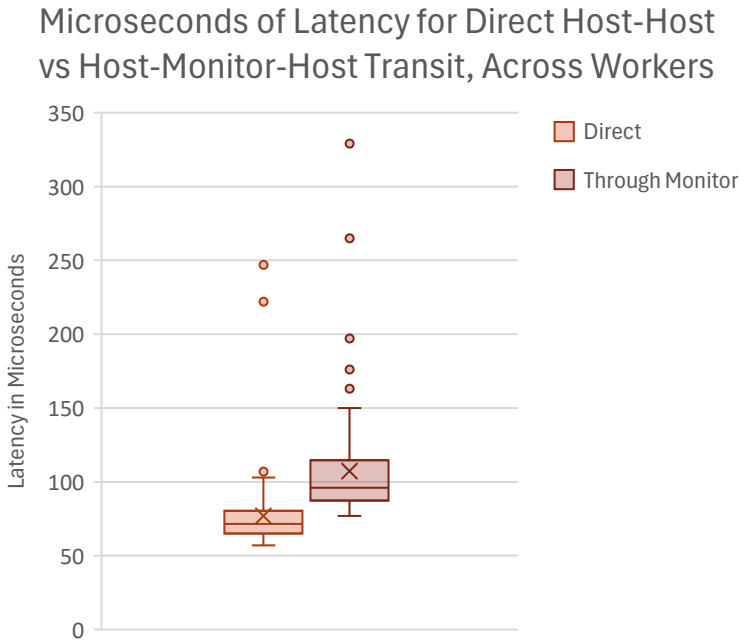
Avg processing overhead: 2.75 μs

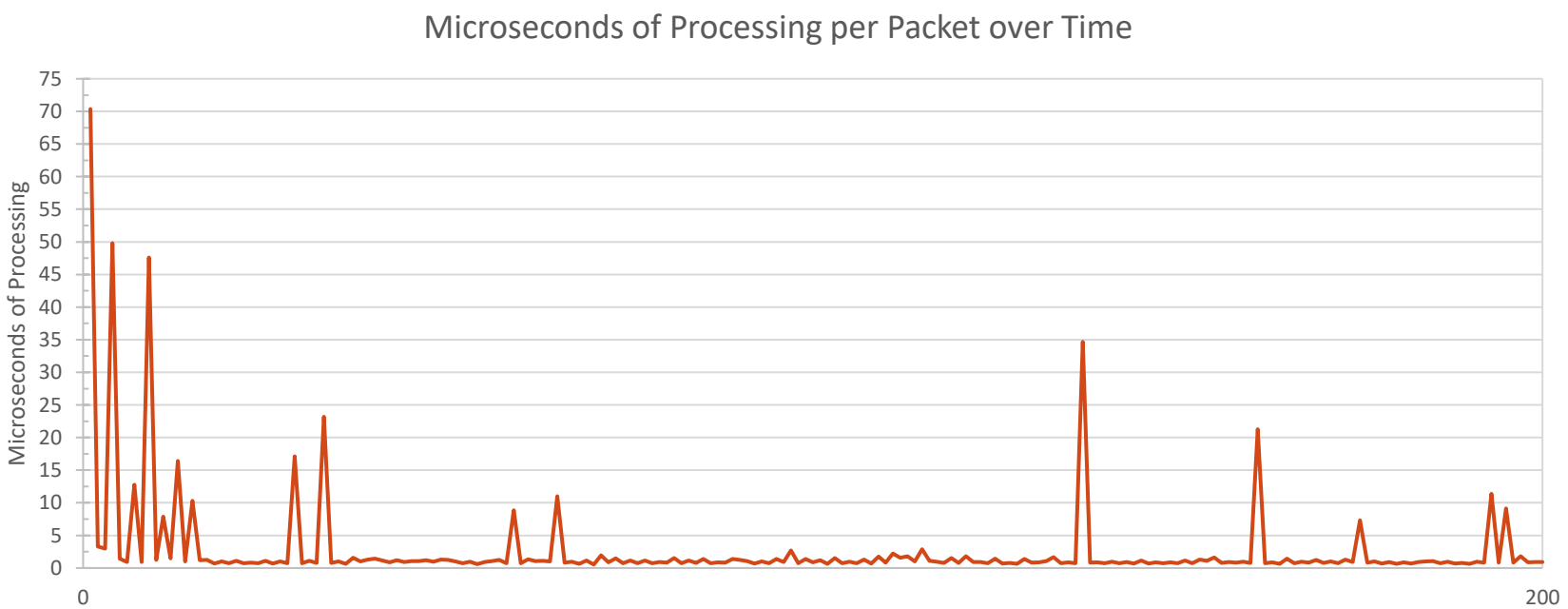


Statistics

Latency increase on adding monitor, Min/Avg/Max: +20/30/82 μs

Average processing overhead: 2.75 μs





Future Work

Trigger based off packet contents

With packet IDs and timestamping, allow for breakpoints and replay

Automation of finding point of failure from packet histories