
✧ Experience ✧

Developing Code for the seL4
Environment





Alain

Computer Science Professor at Lewis and Clark. Project Lead.

Recent Computer Science 2024 Graduate from Lewis and Clark.

Wyeth



Caitlyn

Senior Computer Science major at Lewis and Clark College.

Senior Computer Science & Math major at Lewis and Clark College.

Daniel



Outline

Experience

Environment, Microkit,
sDDF, ipbench





What we are striving for

Verified Software

- seL4 framework and methodology
- Vertically integrated verification

Pure IPv6 Network Stack Implementation

- Separation of concerns
- Support IPv6 and a minimal set of protocols

Competitive Performance

- Latency and bandwidth

How did we get here?

Refresher

- Near completion of all network layers
- Working ping application
- Had recently migrated from Core Platform to Microkit

Milestones

- **Start: 2021**
- **Microkit build: 2023**
- **sDDF build: 2024**
- **Performance: 2024**
- **Release: Soon**



Environment Experience

Lab Environment and Benchmarking

Lab

- Hosts and targets interconnected over a 10Gig network
- Network-connected Power Distribution Unit (PDU) for remote control and automation
- TFTP server for continuous integration workflows of our builds

ipbench

- Ported latency test to IPv6
- Python wrapper scripts developed for batch testing
- Additional ipbench tests in progress, with contributions back to the community
- Hope to continue cleanup as we continue

Microkit and Docker



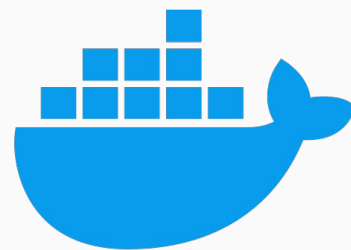
Official Builds

Prebuilt SDKs simplify build environments



Upkeep

Migrating to new versions can be challenging for a small team



Reproducibility

Microkit SDKs can be added to a container in only a few lines

sDDF and Hardware

Boards

- Odroid-C2
- Odroid-C4
- imx8mm

Improvements

- Solidifying APIs
- Growing documentation



Road Bumps

- Benchmarking utilities
- Deconstructing examples

Verification Tools

Isabelle/HOL

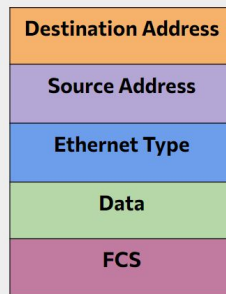
- Writing C because seL4's C parser
- Fragmentation and reassembly
- MAC layer

TLA+

- Queues and Neighbor Discovery
- Inform distributed protocol implementations

Proofing example

```
lemma frame_correct: "VARS (out :: eth_frame)
{True}
out := make_frame data dst src
{src = frame_src out ∧
dst = frame_dst out ∧
data = frame_data out}"
using Valid_def apply fastforce
```



General Experience



Manpower

- Small team
- Cycling members
- Complexities in learning delay starts



Upkeep

- Maturing of seL4's tools
- Staying current
- Official release, API improvements



Community

- Responsive
- Encouraging
- Helpful



★ Performance ★

Let's Visualize

Architecture

Layer Separation

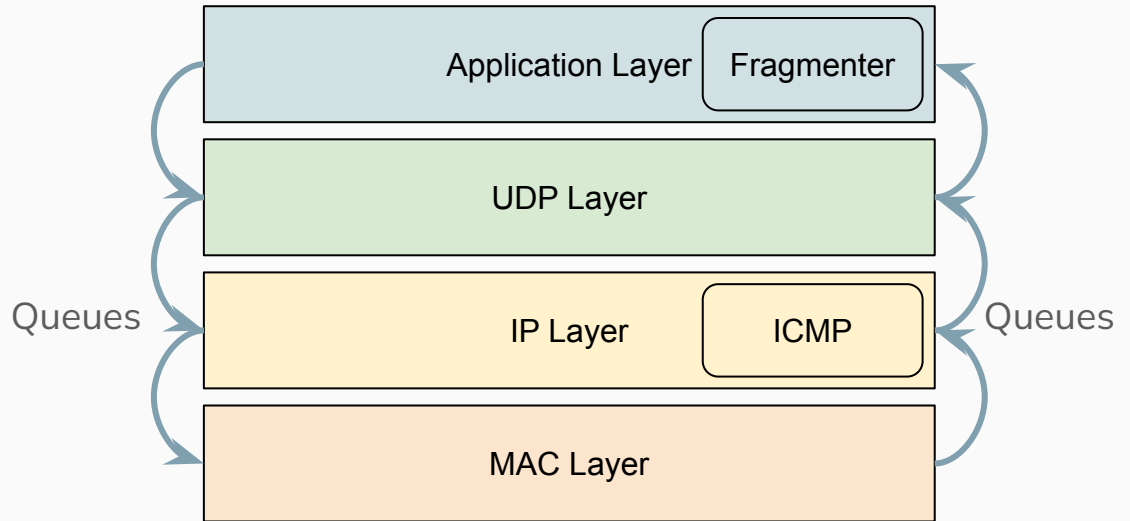
- Promotes modularity and maintainability

Inter-Layer Communication

- Utilizes queues for data exchange
- Aids in consistent layer implementations

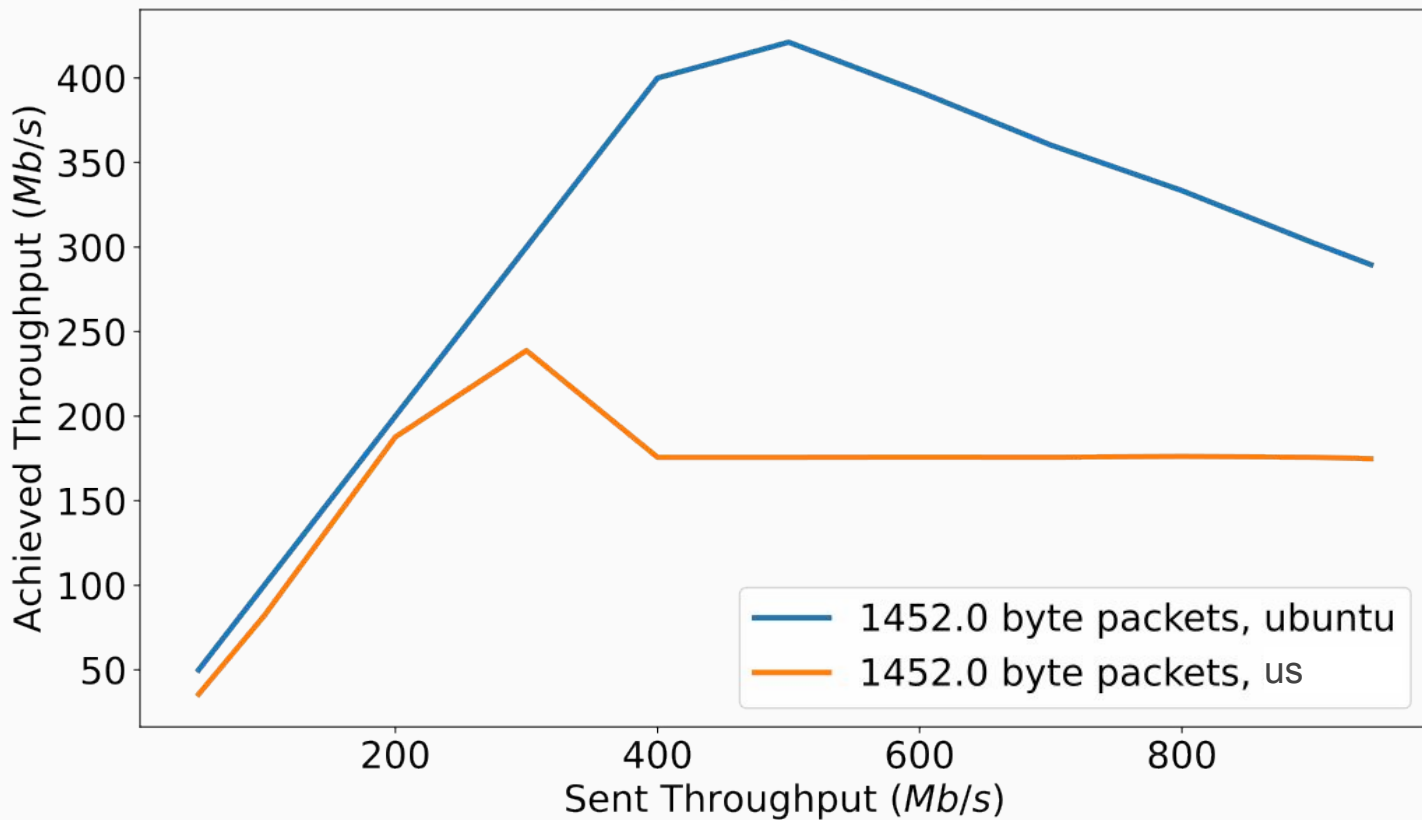
No sDDF Virtualizers

- Decision to minimize signaling overhead
- Targeting systems that don't require complex virtualization



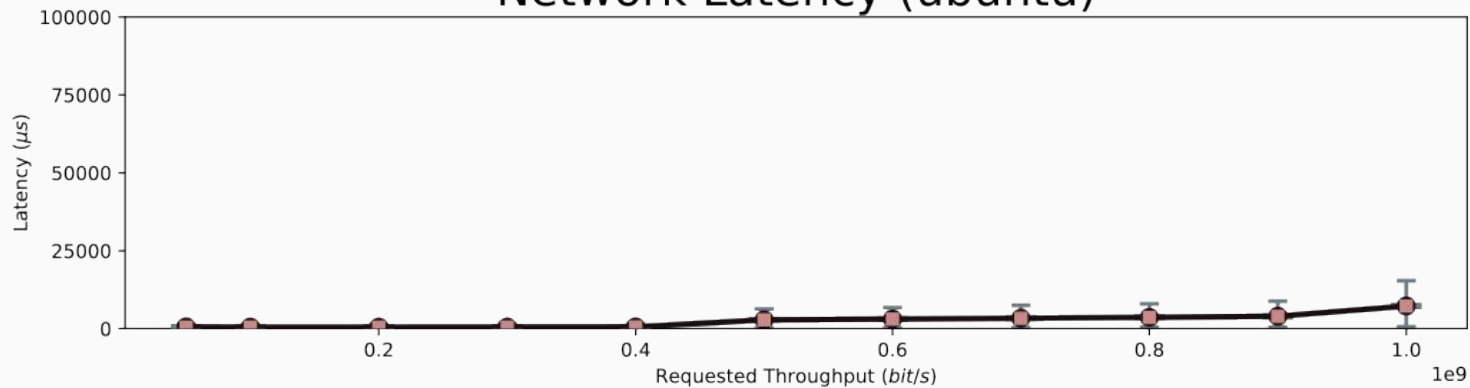


Network Performance

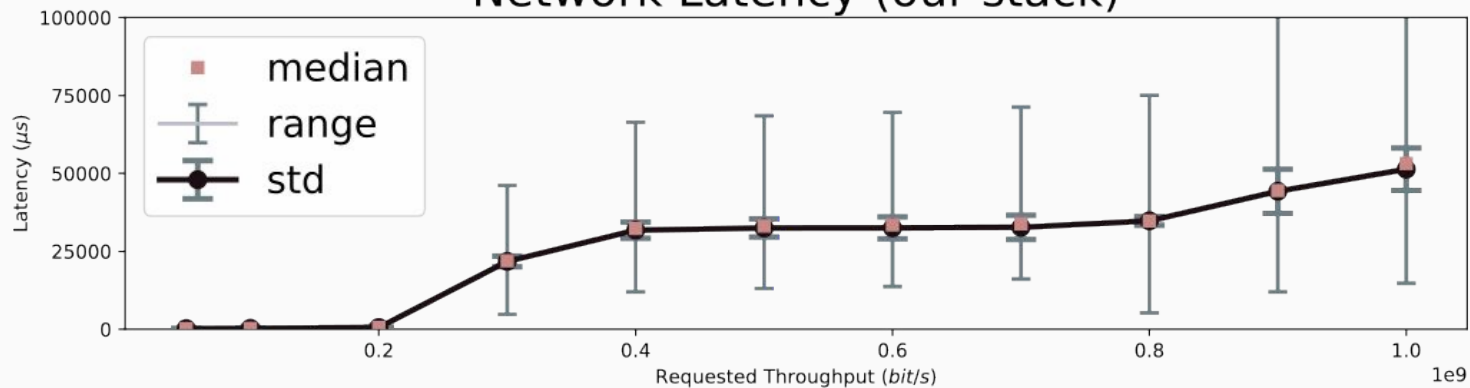




Network Latency (ubuntu)



Network Latency (our stack)





Acknowledgement

We must say thank you to:

The seL4 team:

Birgit Brecknell, Matthew Brecknell, Peter Chubb, Gernot Heiser, Gerwin Klein,
Tessa Lunney, Lucy Parker, Johannes Åman Pohjohla, Ivan Velickovic

Evergreen State College:

Richard Weiss,
Aubrey Birdwell, Seth Leichsenring, Levi Overcast

Lewis & Clark College:

Jens Mache,
Dylan Angel, Linus Brogan, Christian Ermann, Michael Harper, Syd Parker



Thanks!



Do you have any questions?

Contact the team here:

alaink@lclark.edu

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)
