School of Computer Science & Engineering

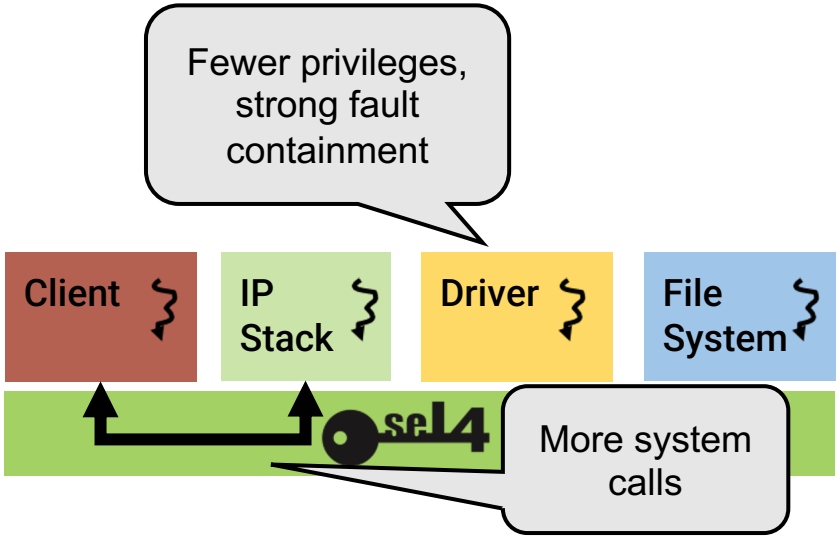
Trustworthy Systems Group

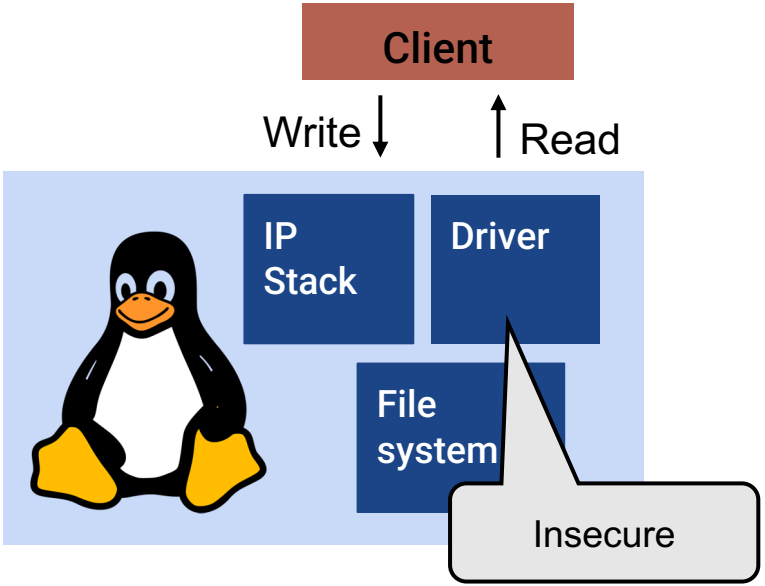
The seL4 Device Driver Framework

Lucy Parker
lucy.parker@student.unsw.edu.au

The seL4 Device Driver Framework

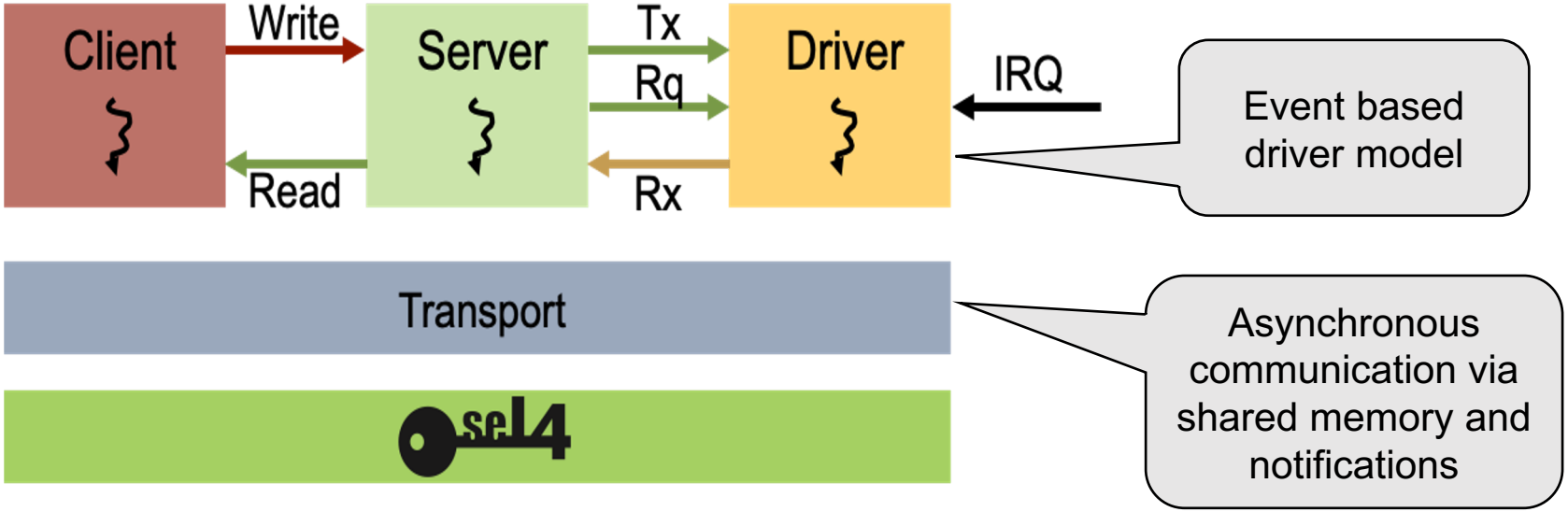
Framework to provide interfaces and protocol for writing performant device drivers as seL4 user level programs.





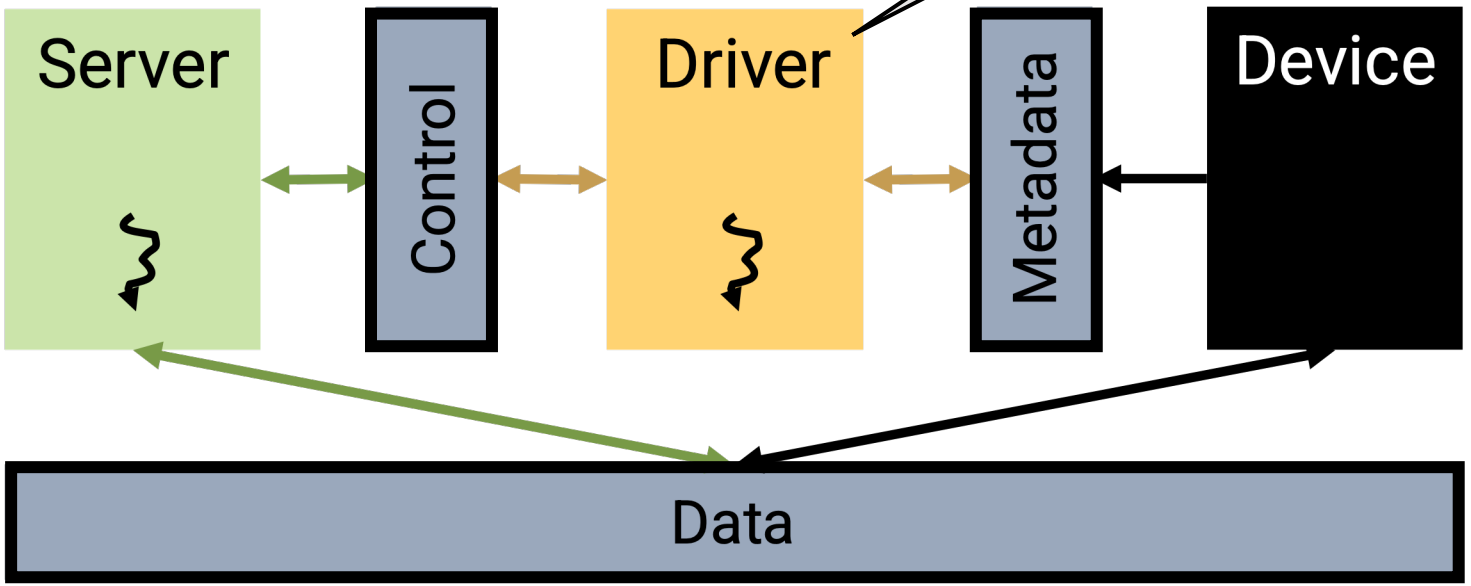
What Is The sDDF?

- Currently focused on networking systems
- Implemented on top of the seL4 Microkit



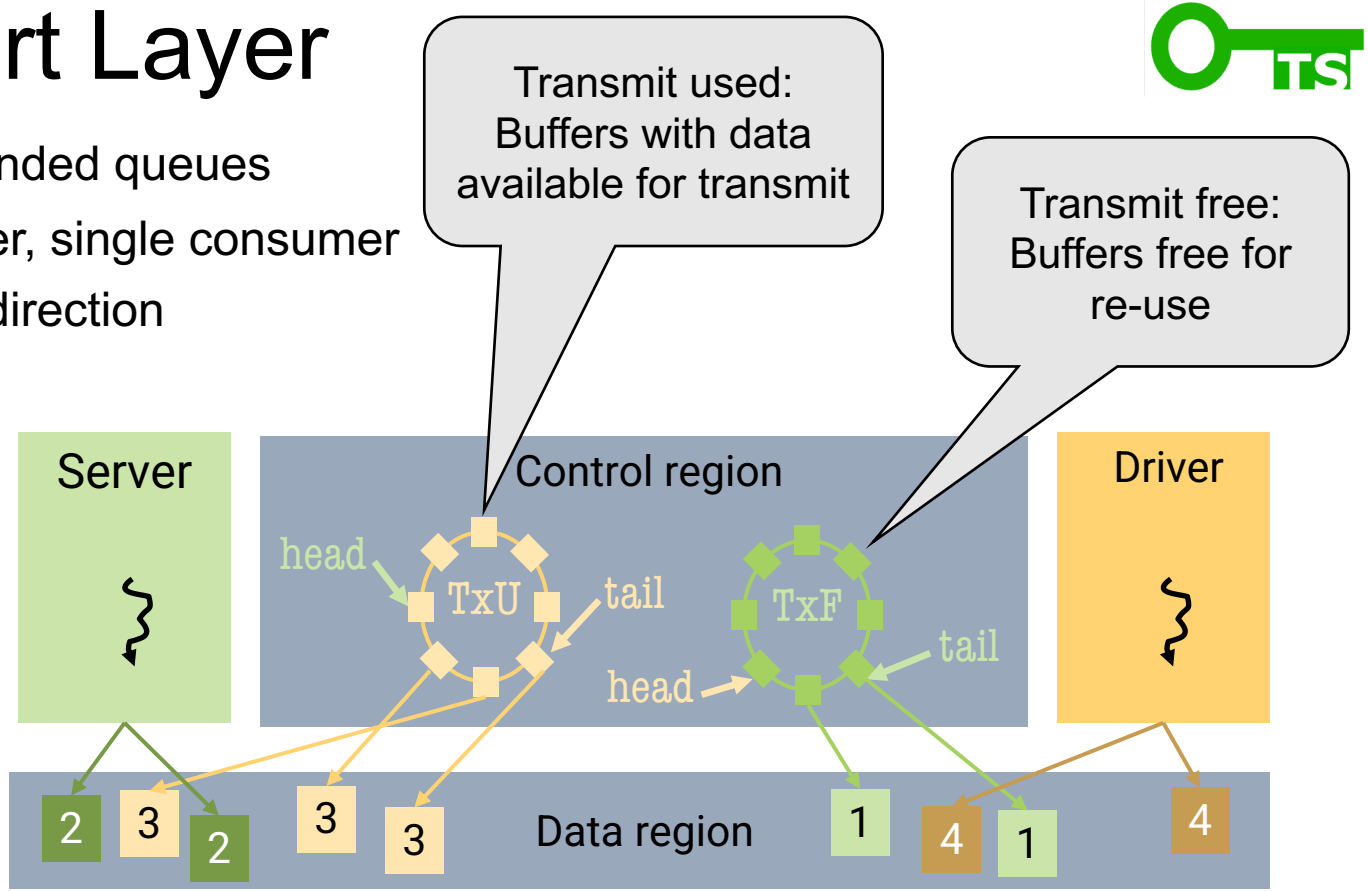
Design

- Driver model uses 3 different memory regions
- Notifications signal updates to these regions

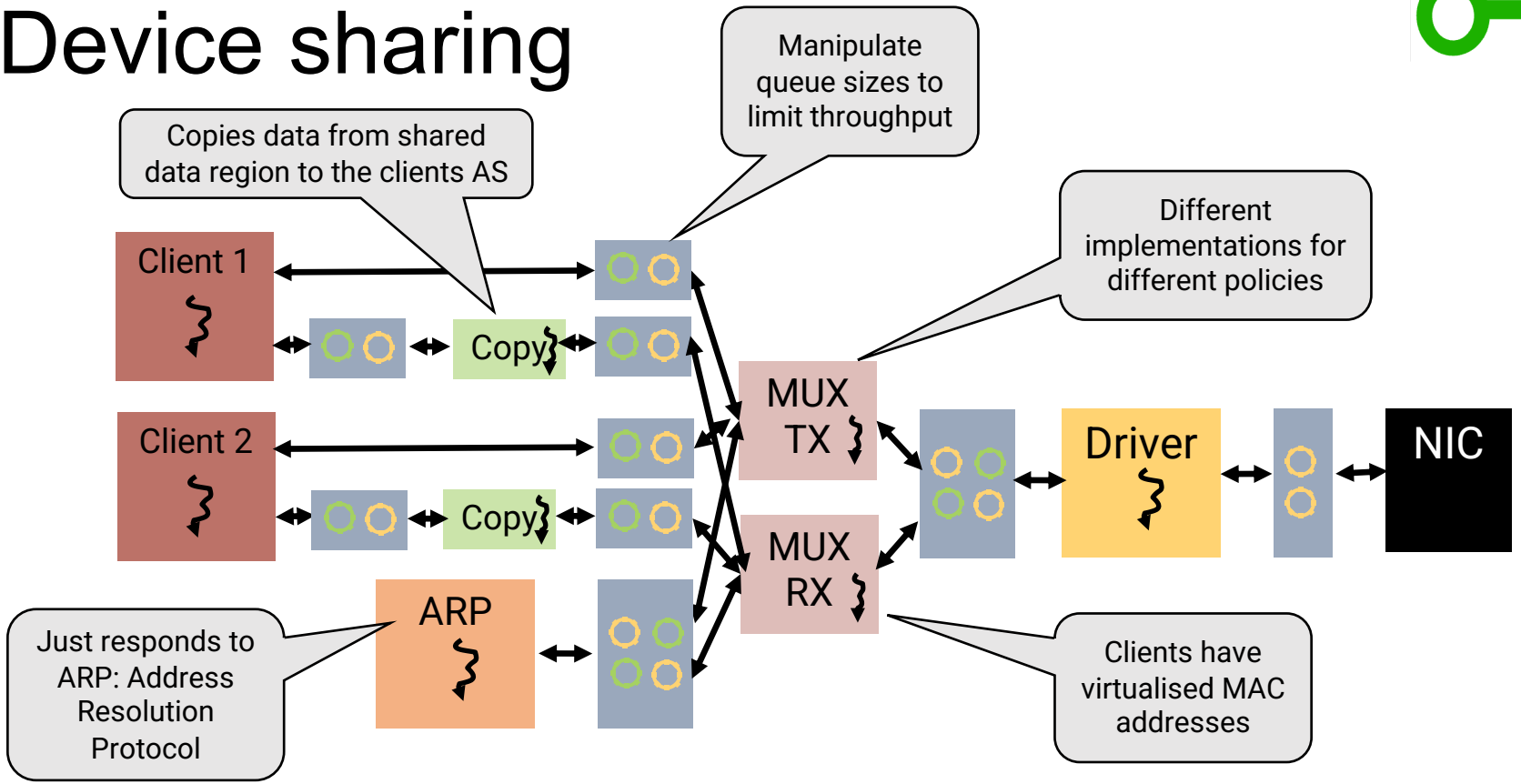


Transport Layer

- Lock free, bounded queues
- Single producer, single consumer
- 2 queues per direction
- Zero copy



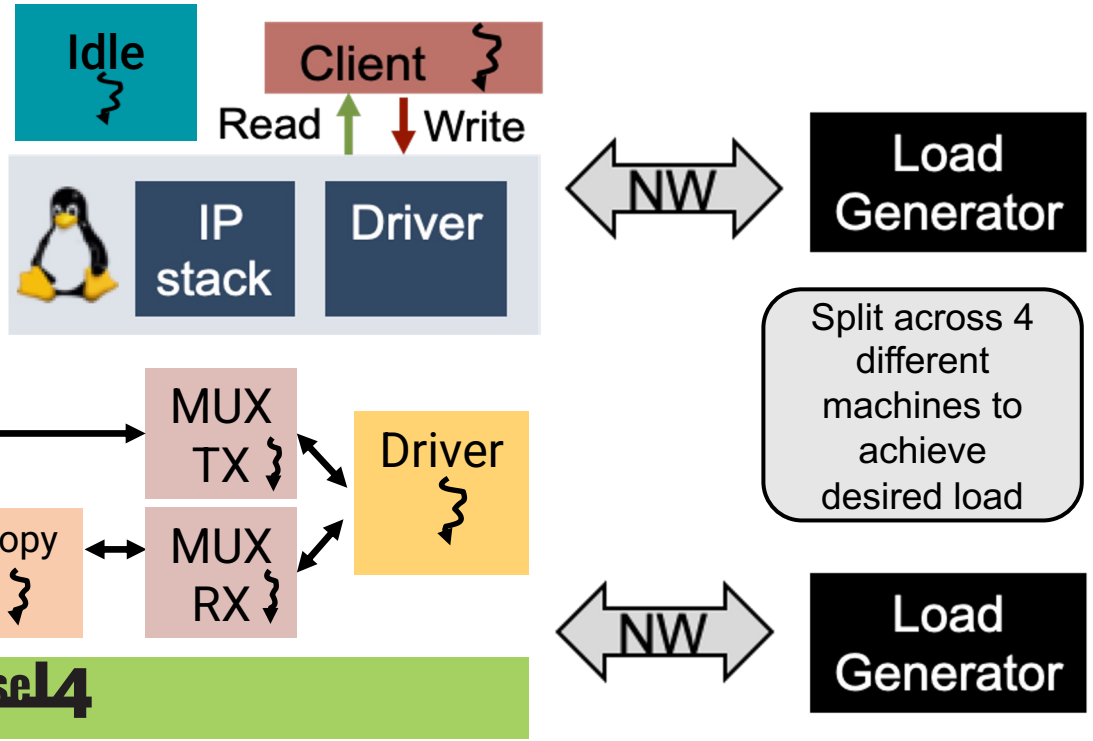
Device sharing



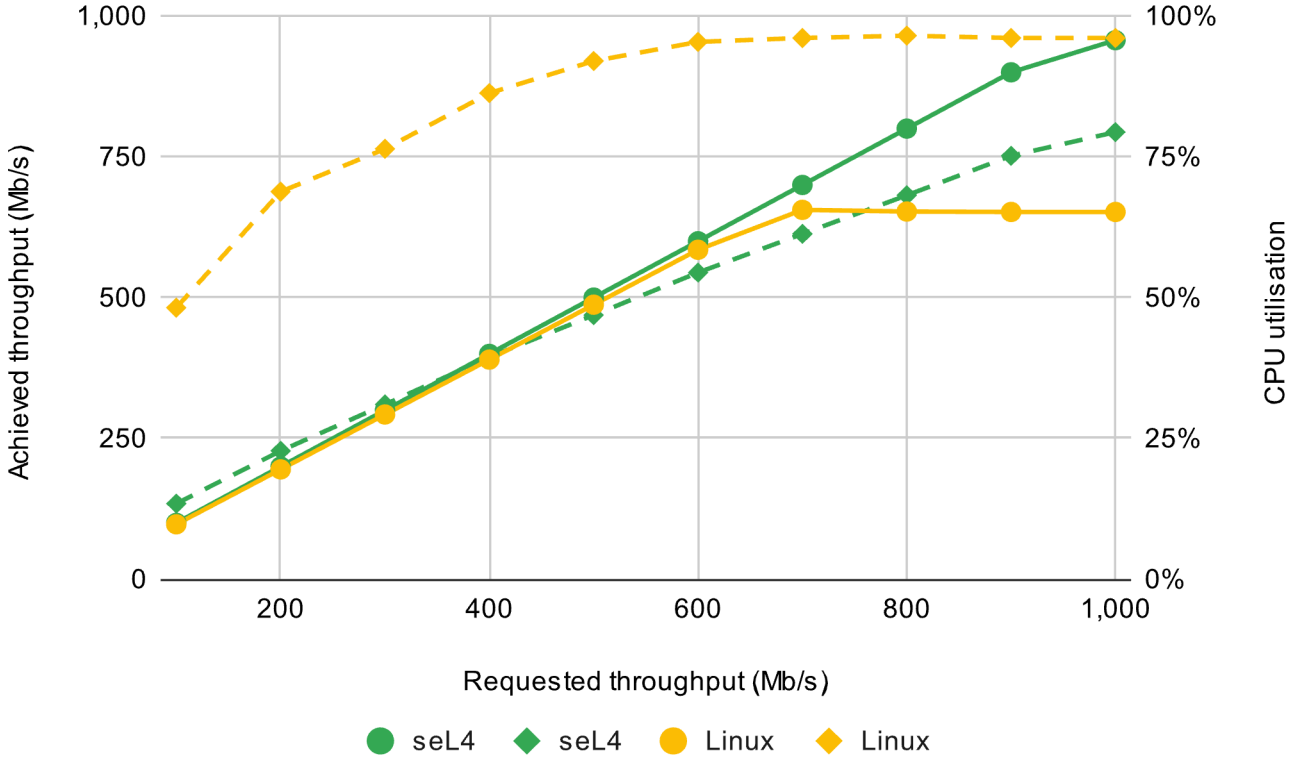
Performance

Set up

- Client just echoes packets
- IPbench sends UDP packets and measures throughput and latencies
- Idle thread counts cycles to calculate CPU utilisation



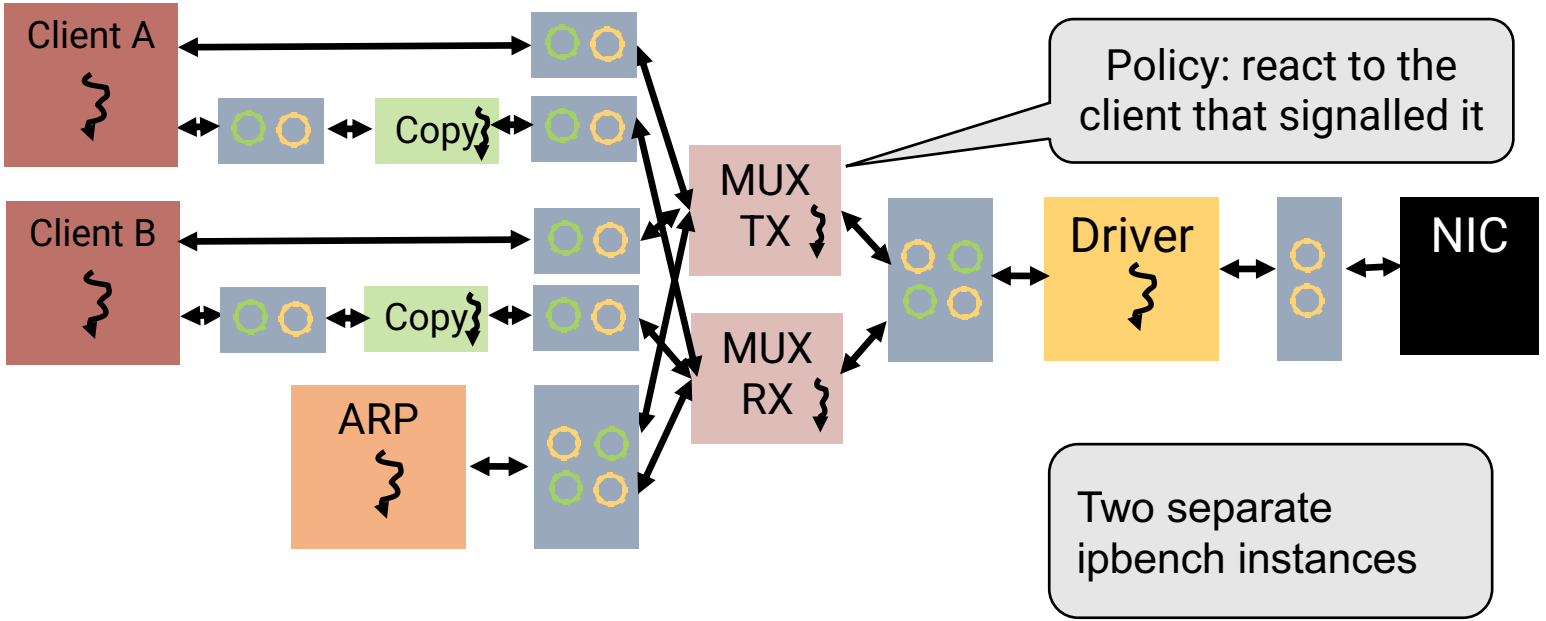
sDDF vs Linux



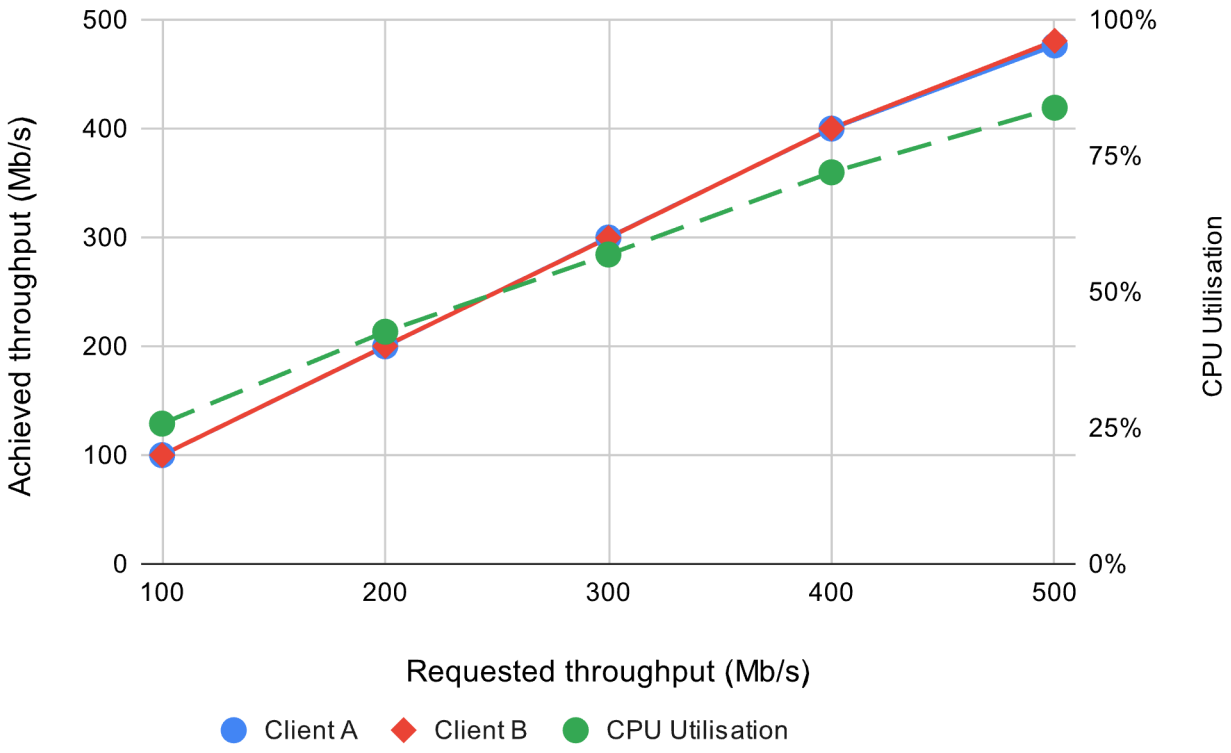
Multi-client Performance

Set up

Priorities:
Driver > Tx Mux > Rx Mux >
Copier A, Copier B > Client A, Client B



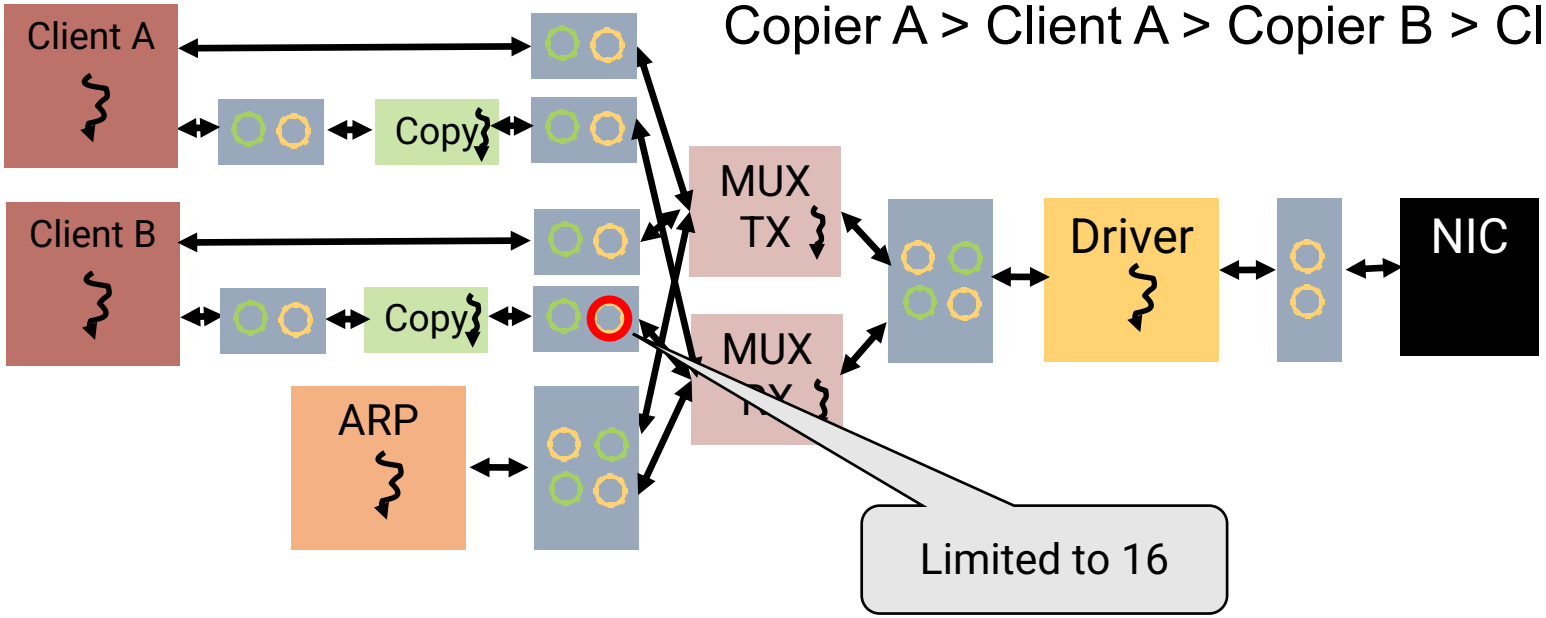
Performance



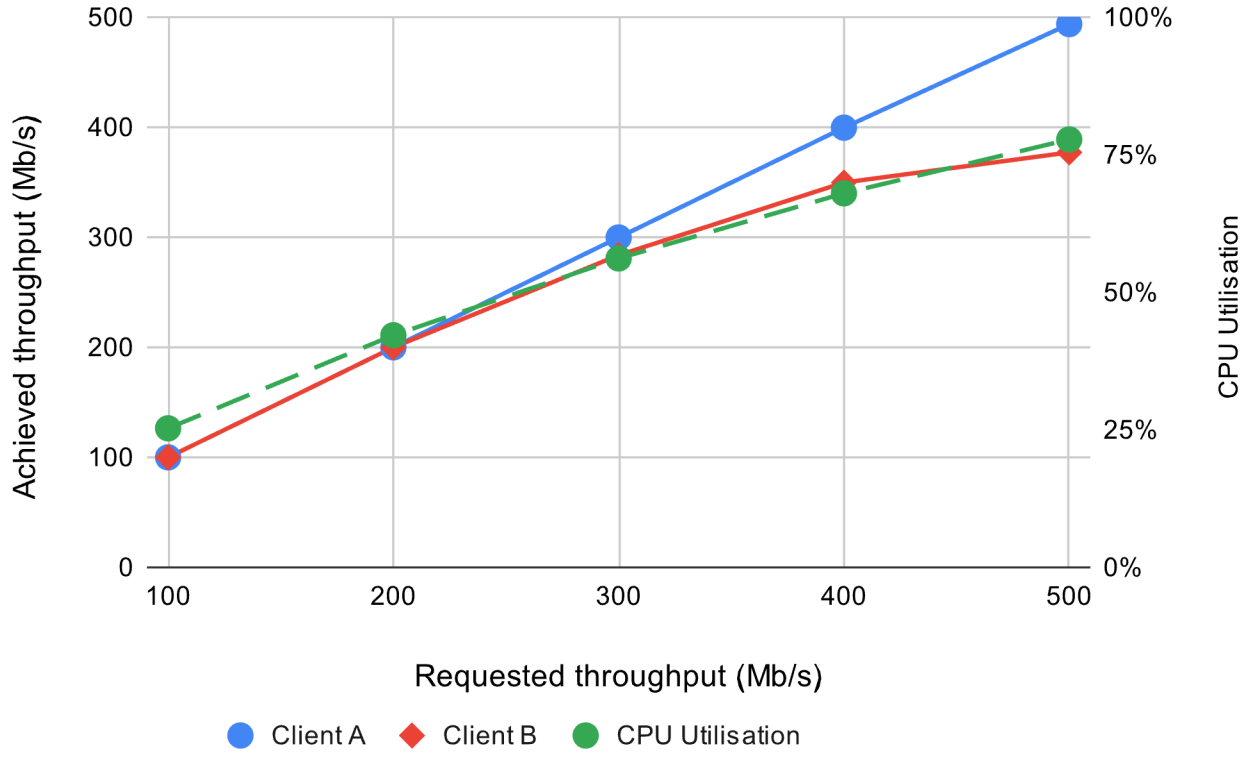
Echo servers:
equal priorities
equal queue sizes

Set up

Priorities:
Driver > Tx Mux > Rx Mux >
Copier A > Client A > Copier B > Client B



Performance



Echo servers:

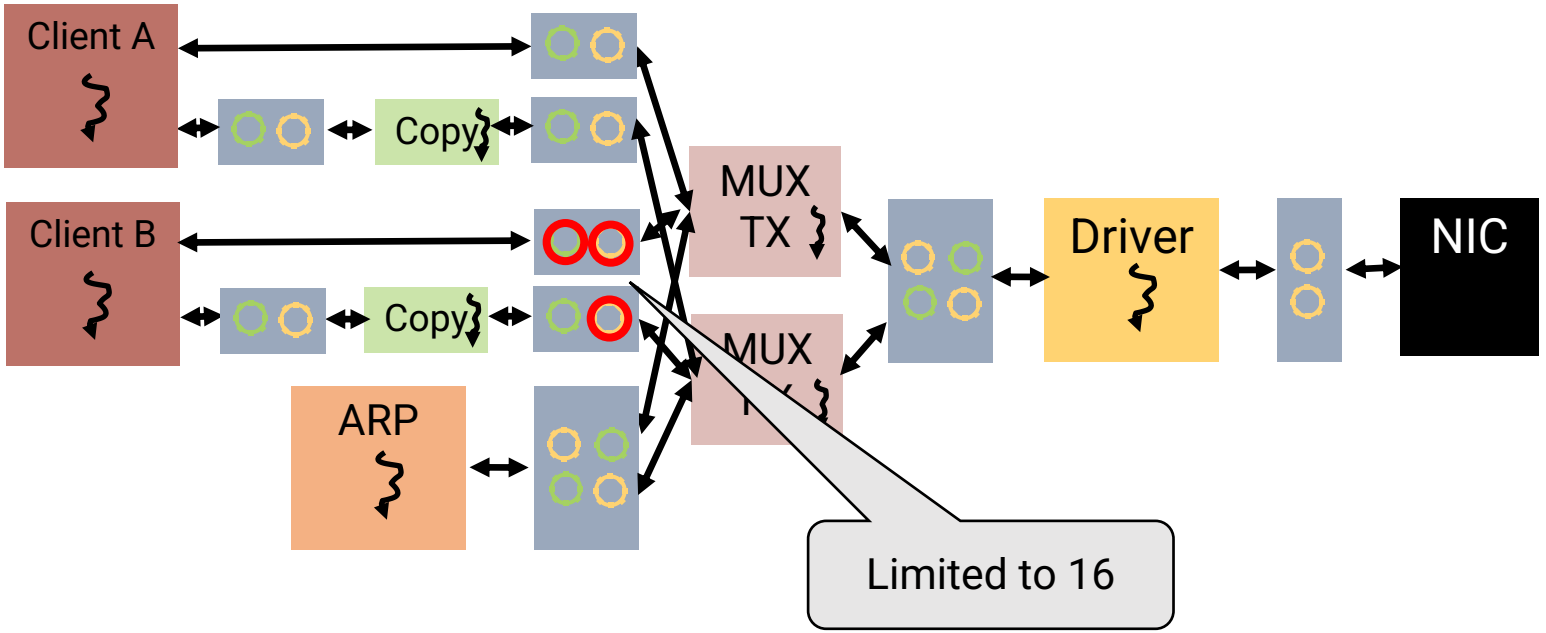
Client A > Client B

Client B RxUQ: 16

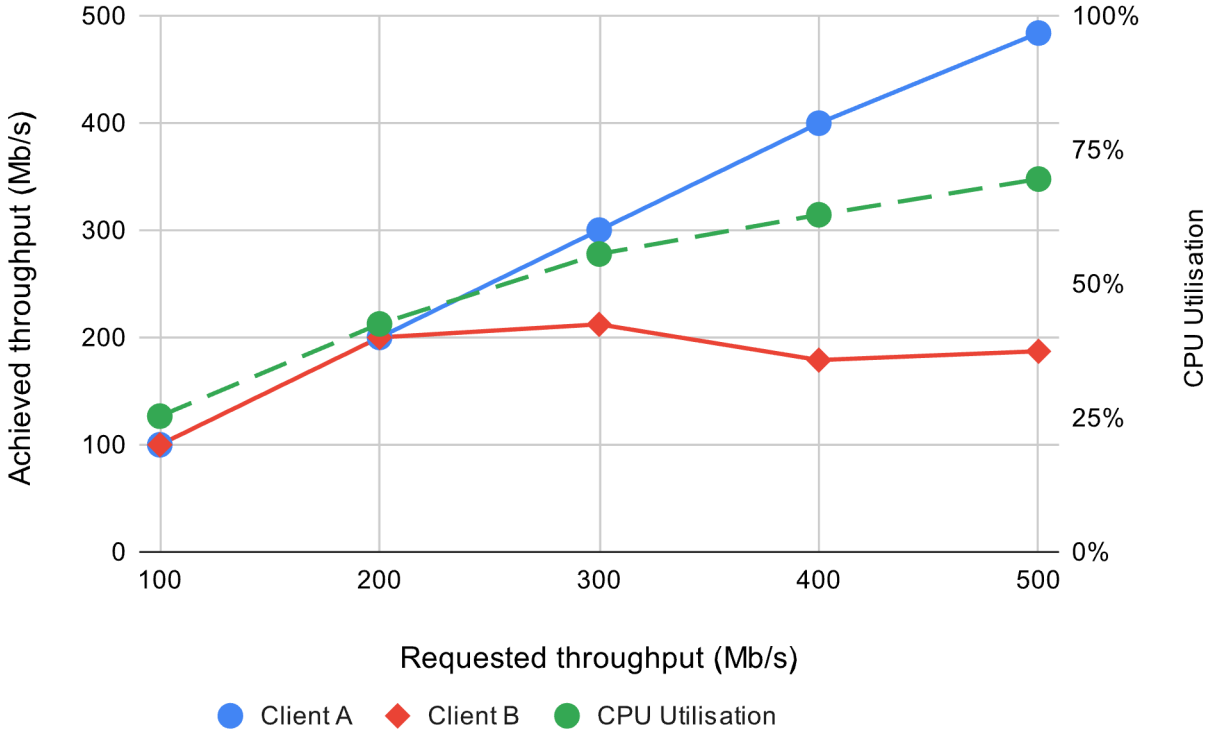
The same multiplexer!

Set up

Priorities:
Driver > Tx Mux > Rx Mux >
Copier A > Client A > Copier B > Client B



Performance



Echo servers:

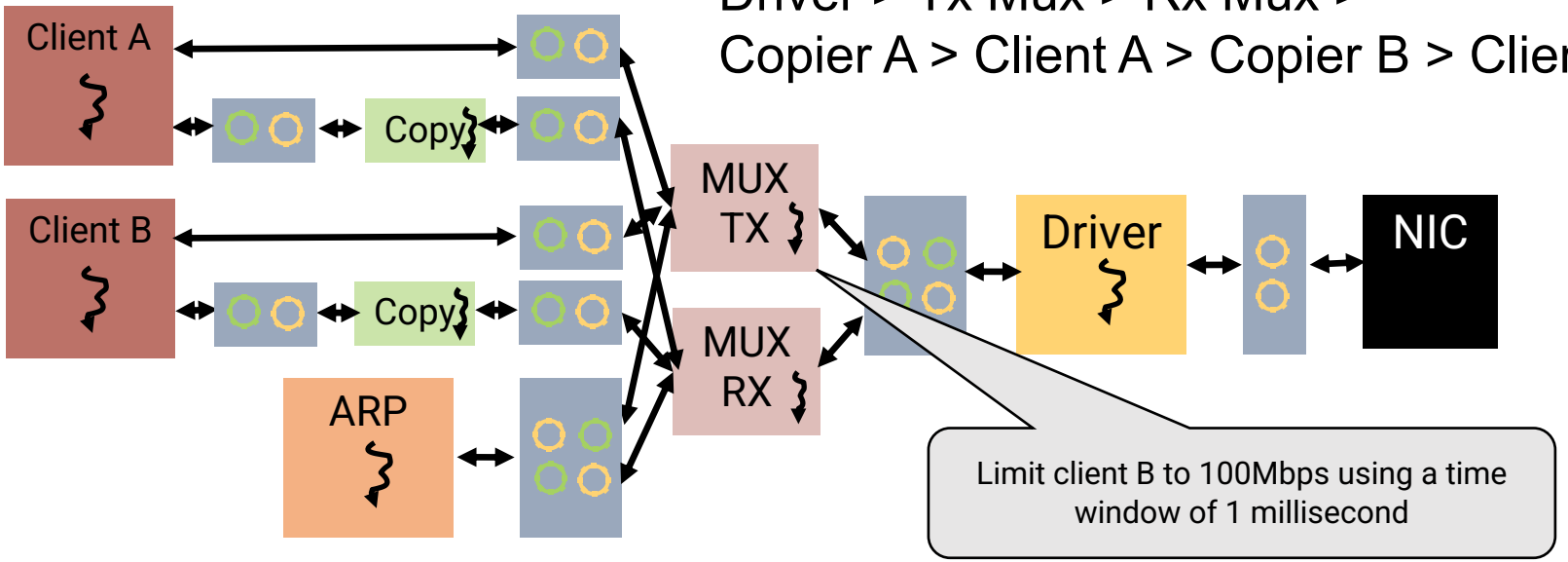
Client A > Client B

Client B RxUQ: 16
TxUQ: 16, TxFQ: 16

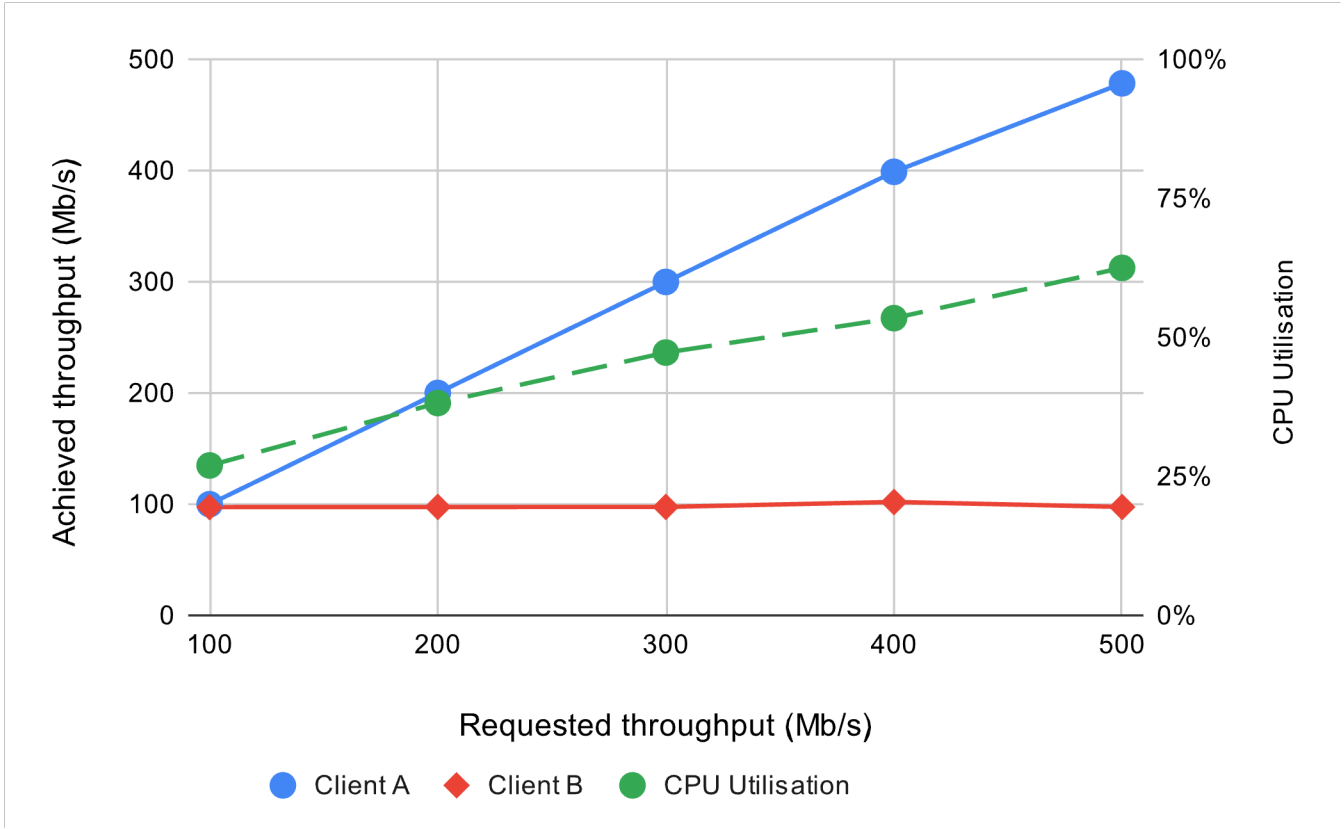
The same multiplexer

Set up

Priorities:
Driver > Tx Mux > Rx Mux >
Copier A > Client A > Copier B > Client B

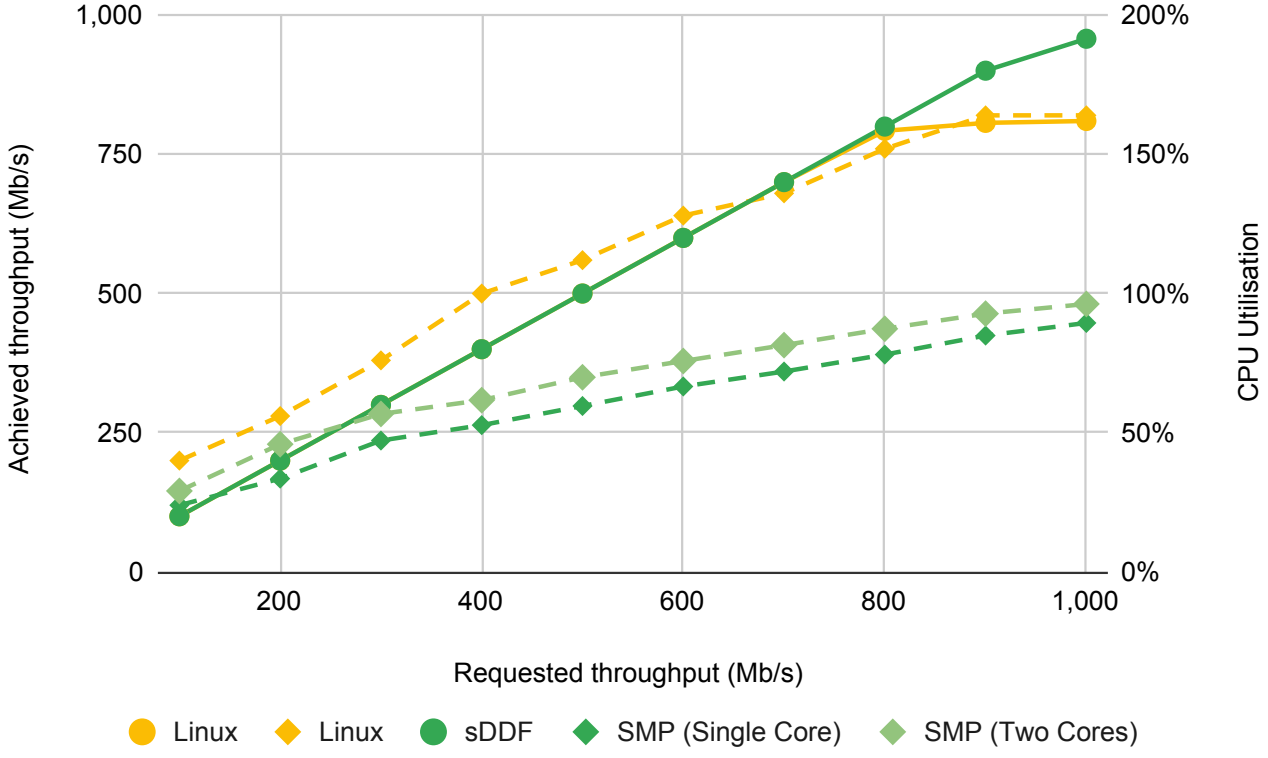


Performance

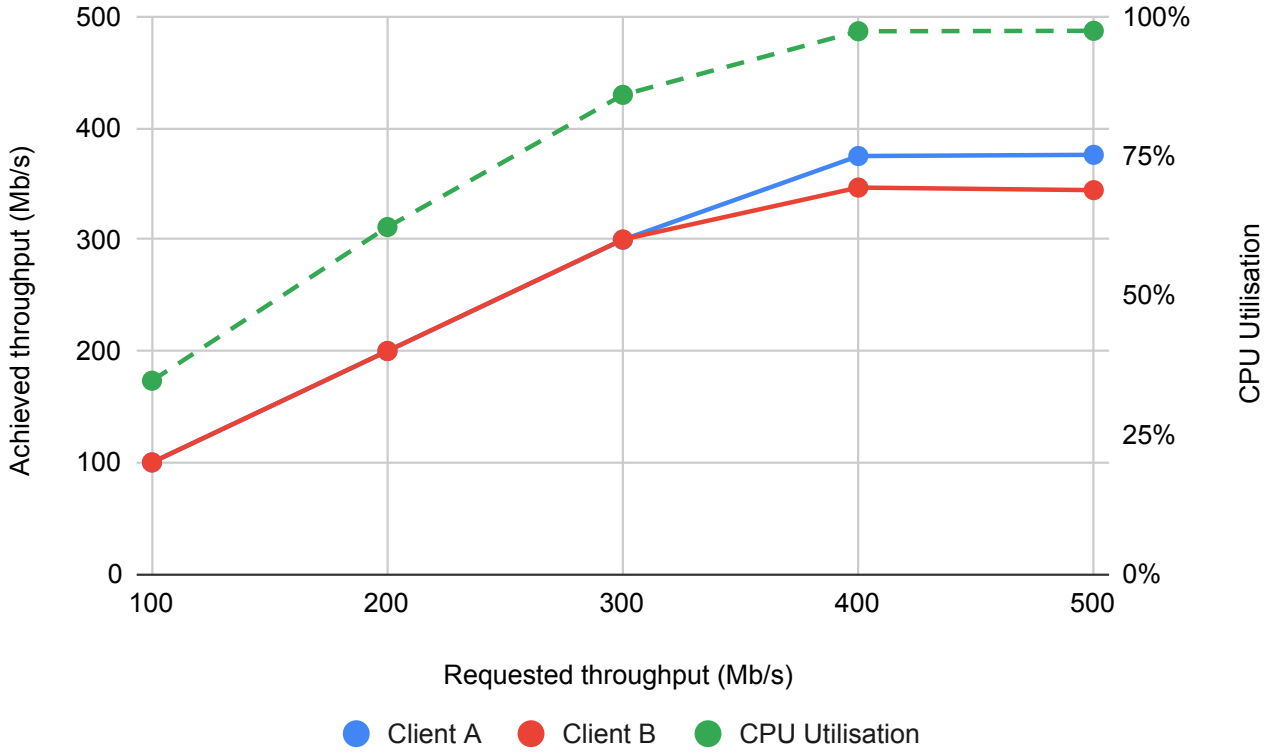


Multi-core

Performance



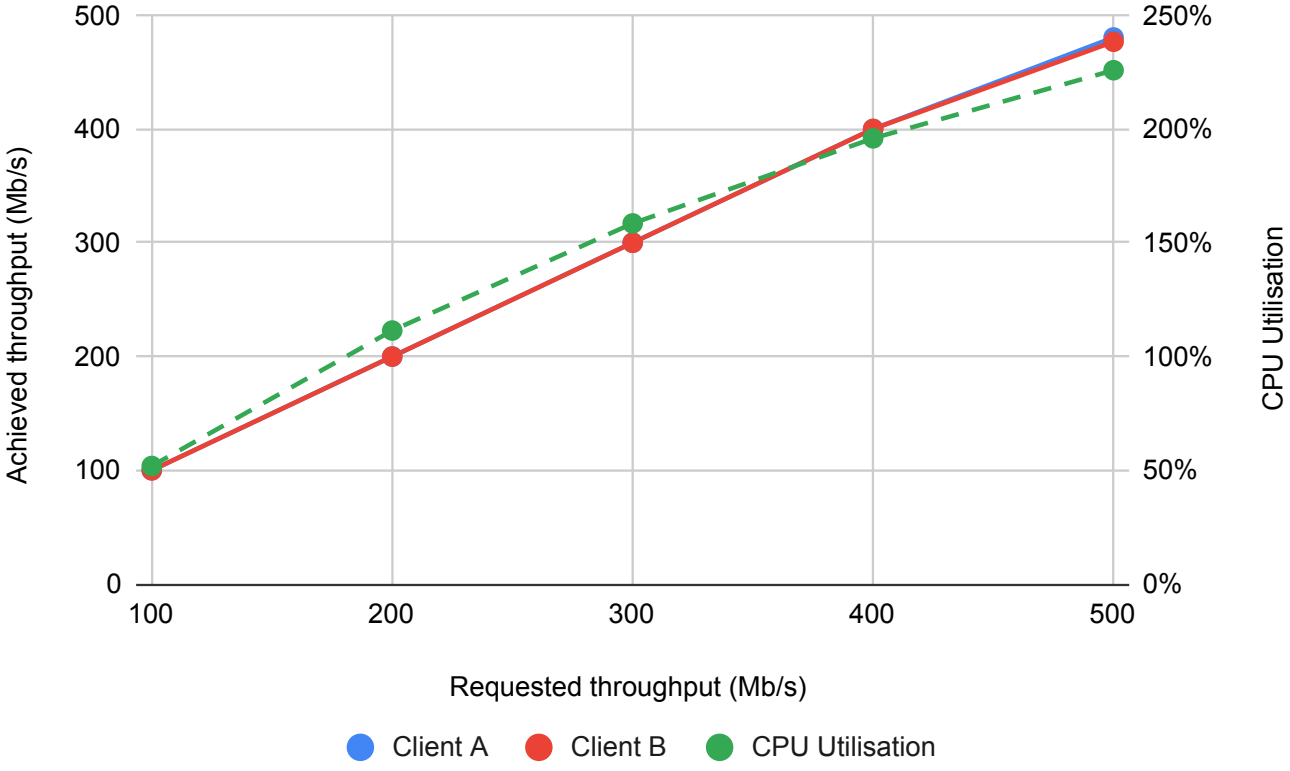
Multi-client, Single Core



Echo servers with extra work

Equal priorities

Multi-client, Multi-core



Echo servers with extra work

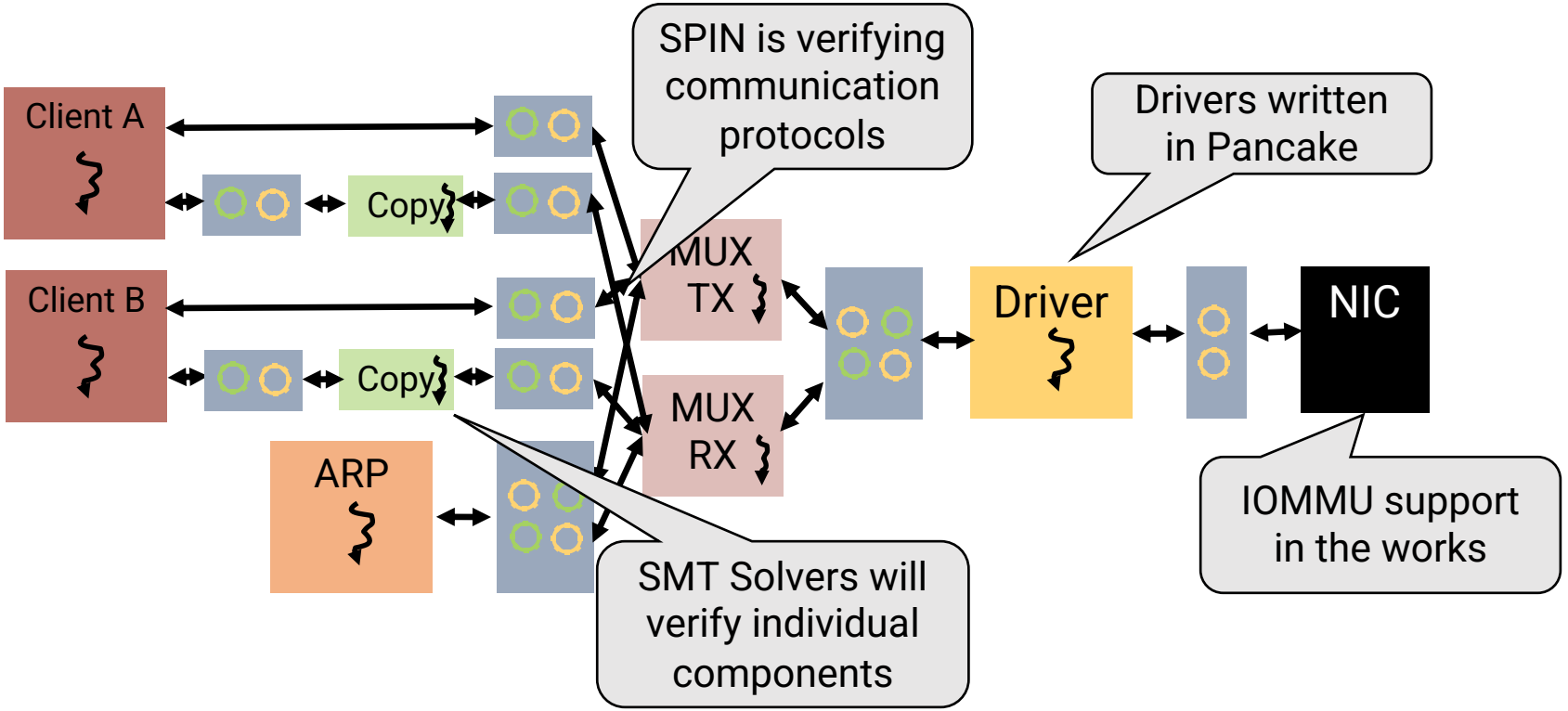
Equal priorities,

Separate cores.

Round robin TX MUX

So it performs… but how do we know it works?

Verification story



Takeaways

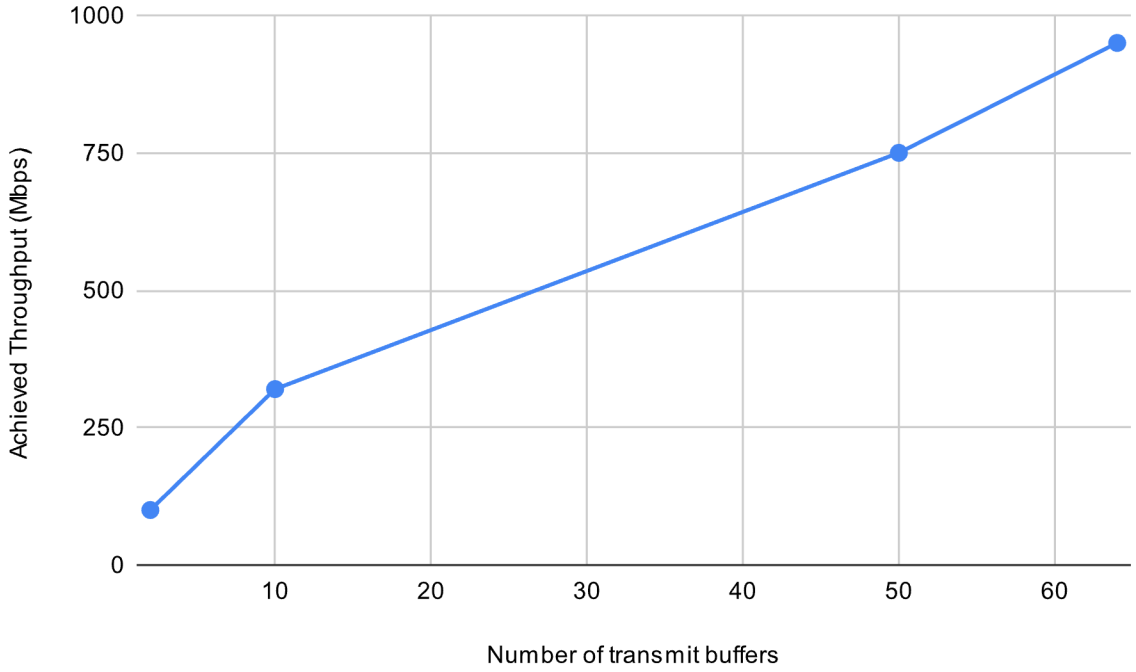
- Simple design outperforms Linux
- Simple, isolated multiplexers do not impede performance
- Inner policies depend on greater system design
- We can manipulate queue sizes to implement different policies

Further Work

- Multi-core performance needs further investigation.
- Could we outperform Linux user-space frameworks too?
- Currently working on other device classes: storage, i2c, driver VMs etc.
- Investigate IOMMU/SMMU support
- Verification

Thank you

Limiting Tx Queues (single client)



Single Core Round Trip Times