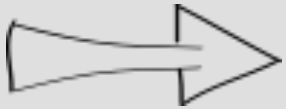
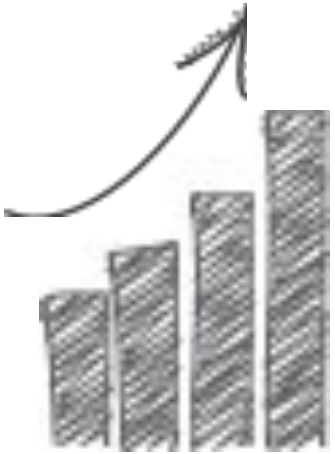


Open-NFP Summer Webinar Series: CAANS: Consensus As A Network Service

Huynh Tu Dang, Daniele Sciascia, Pietro Bressana,
Han Wang, Ki Suh Lee, Hakim Weatherspoon,
Marco Canini, Fernando Pedone, and Robert Soulé
Università della Svizzera italiana (USI),
Cornell University, and Université catholique de Louvain

Outline

- Introduction
- CAANS
- Demo
- Results
- Conclusion



Introduction

Many distributed problems can be reduced to consensus

- E.g., Atomic commit, leader election, atomic broadcast

Consensus protocols are the foundation for fault-tolerant systems

- E.g., OpenReplica, Ceph, Chubby, etc.,

Paxos: one of the most widely used consensus protocols

- Paxos is **slow**
- Optimizations: Generalized Paxos, Fast Paxos

Motivations

Network devices are becoming:

- More powerful: NFP-6xxxx, PISA chips, FlexPipe
- More programmable: custom pipeline processing

Motivations

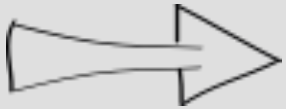
Network devices are becoming:

- More powerful: NFP-6xxxx, PISA chips, FlexPipe
- More programmable: custom pipeline processing

High level languages:

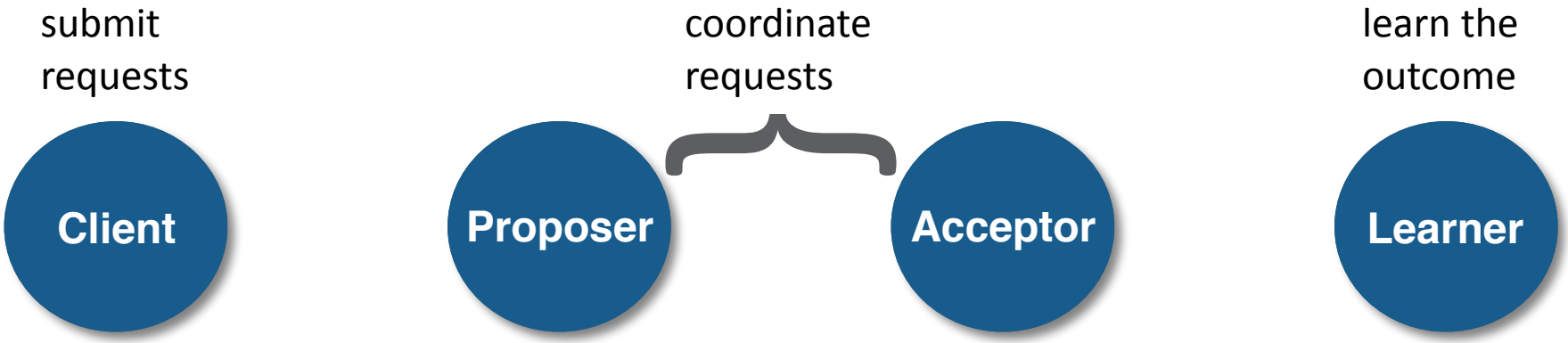
- OpenFlow, PX, P4

Co-design networks and consensus protocols

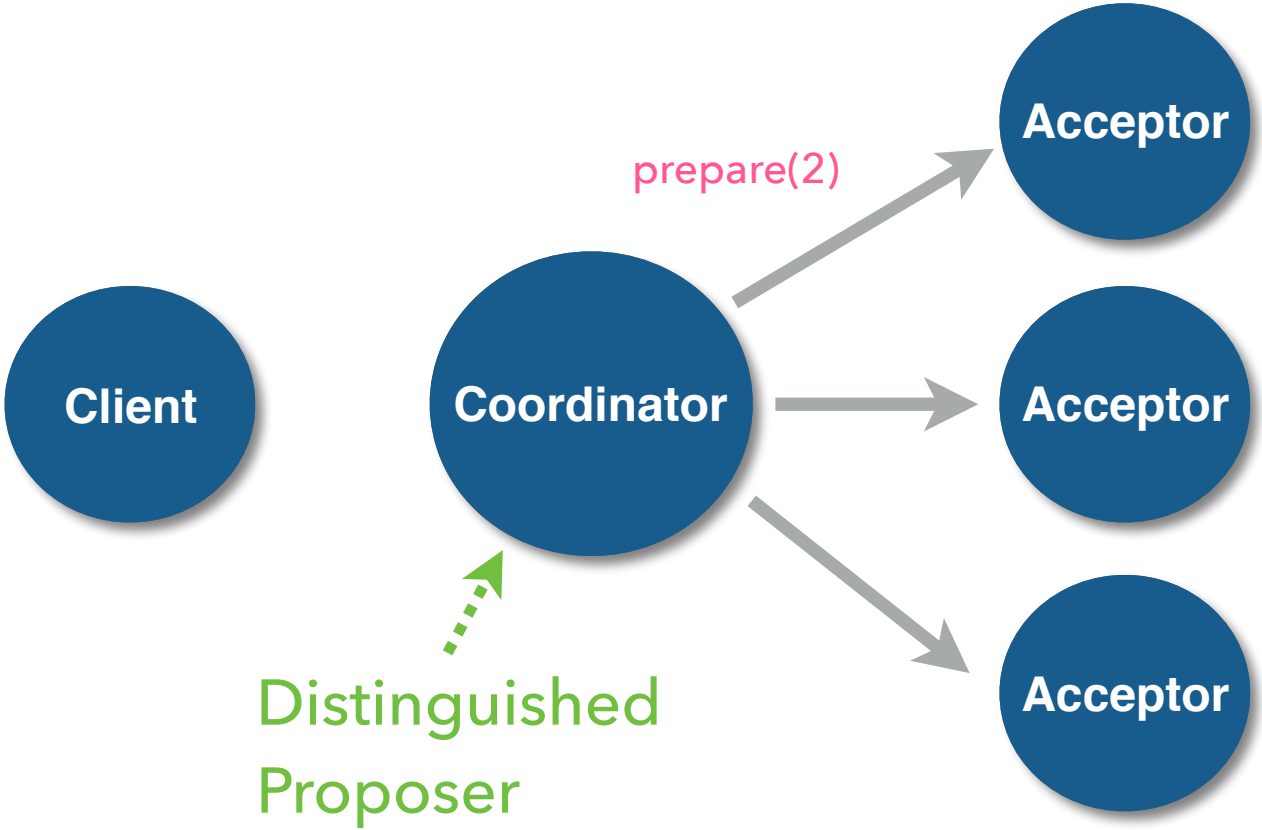


Background: Consensus Problem

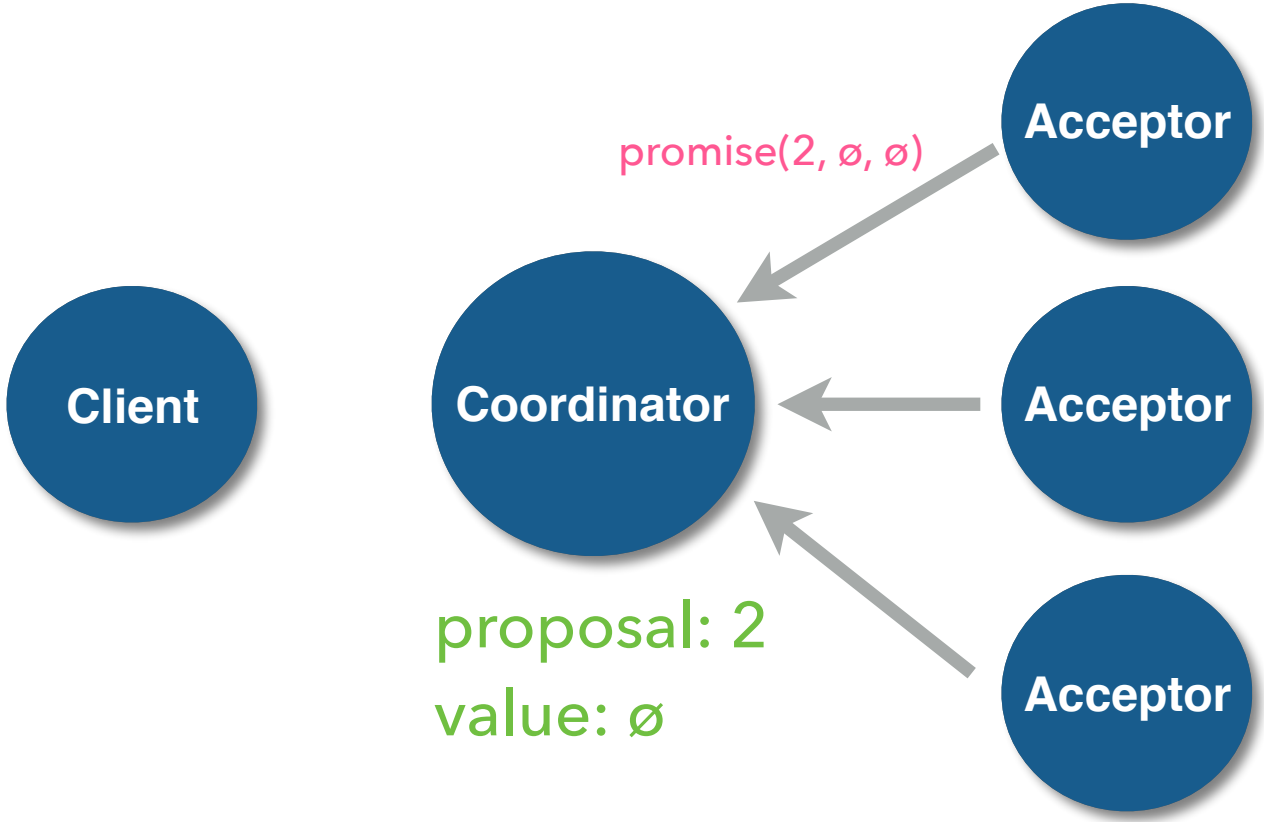
Background on Paxos Consensus



Background on Paxos Consensus

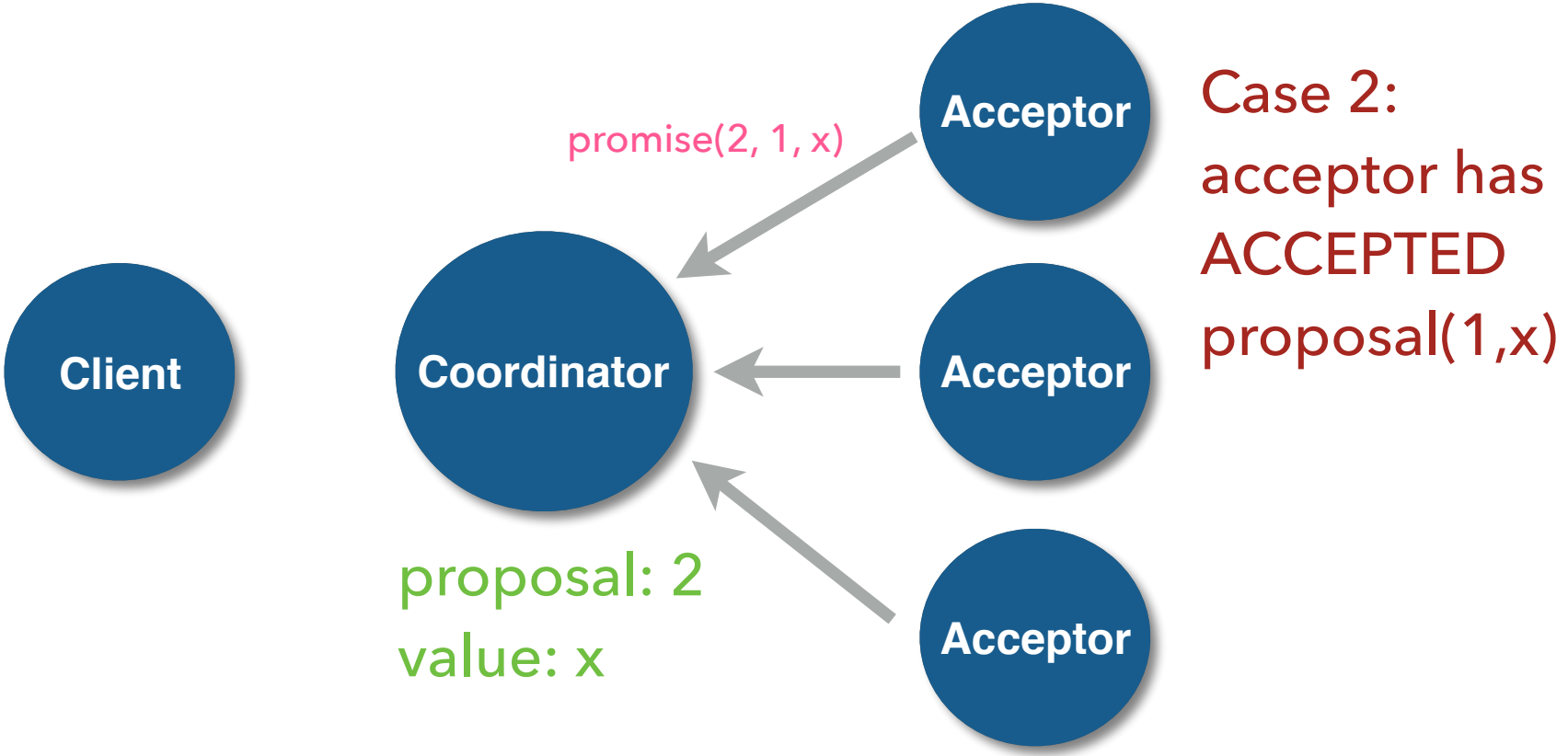


Background on Paxos Consensus



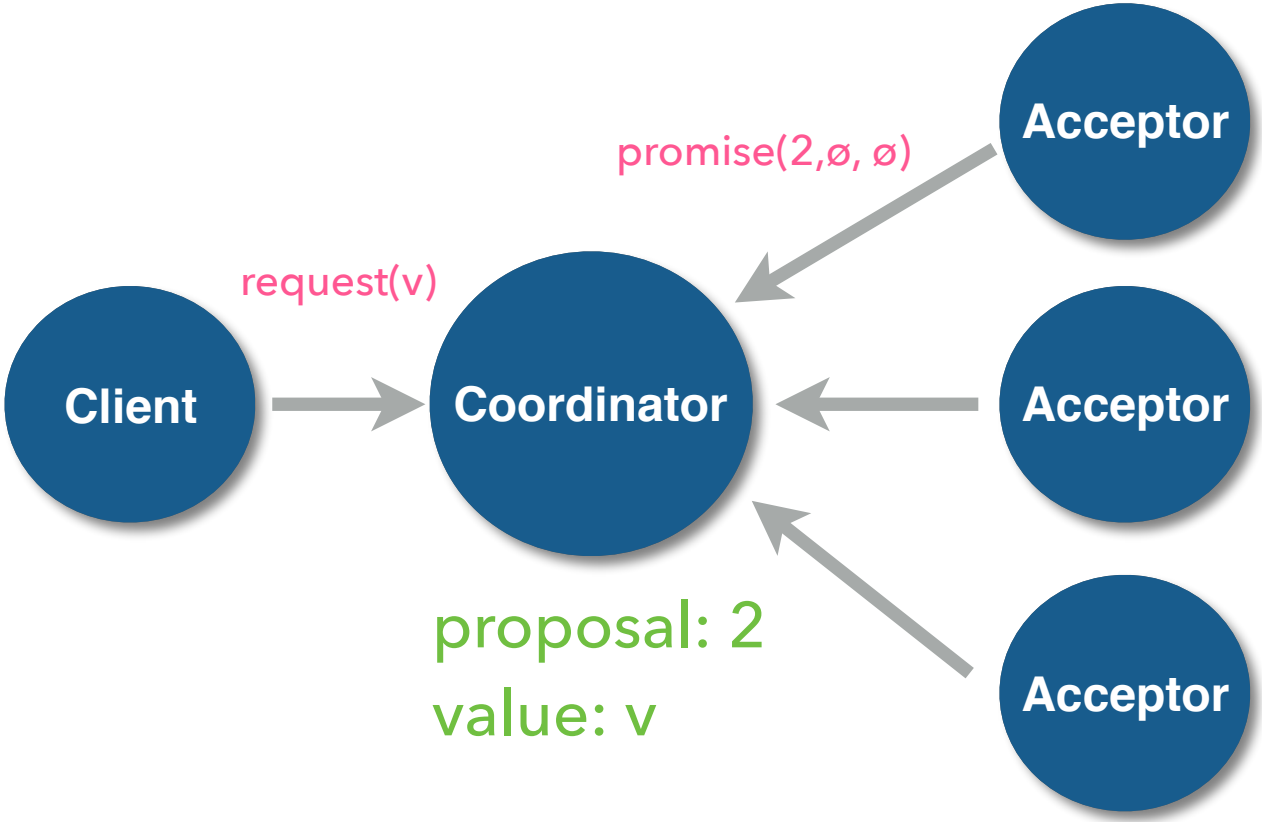
Case 1: acceptor has NOT accepted any message previously

Background on Paxos Consensus

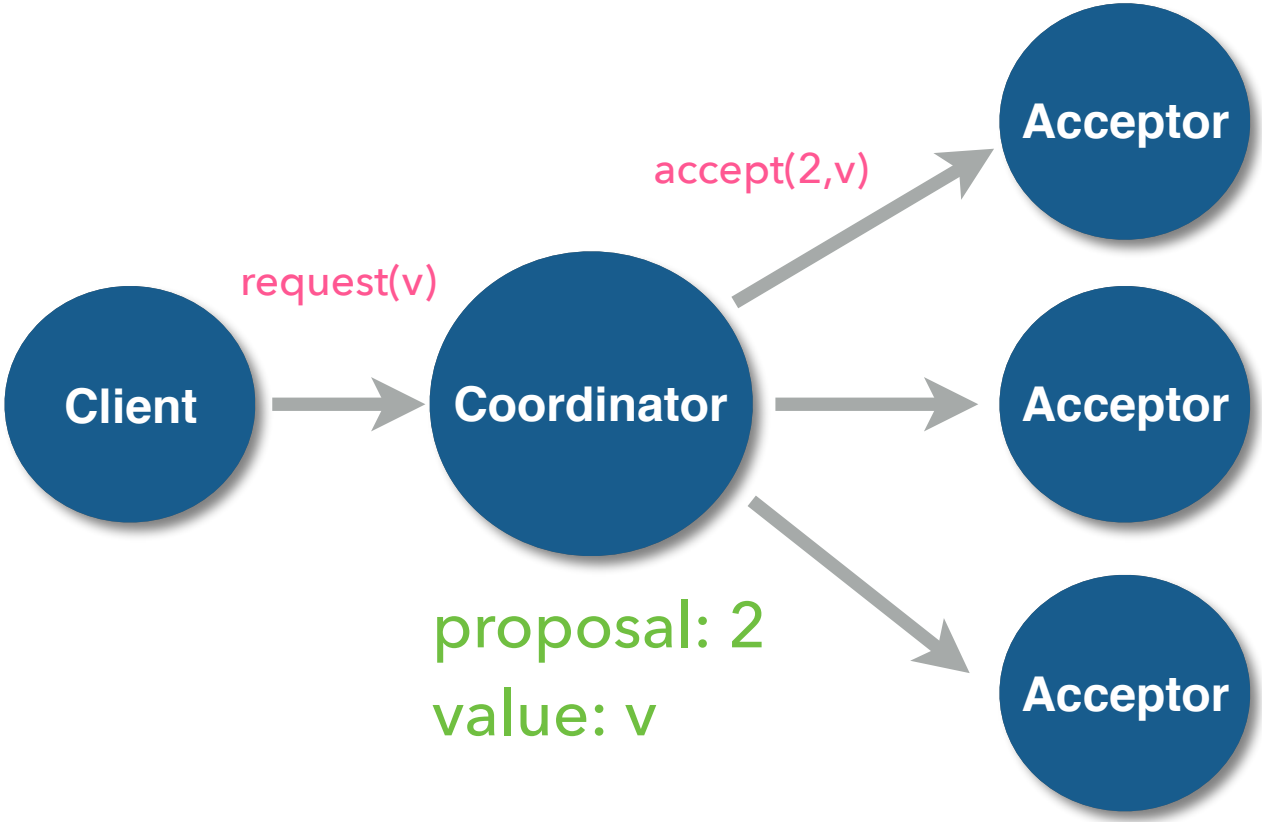


Case 2:
acceptor has
ACCEPTED
proposal(1,x)

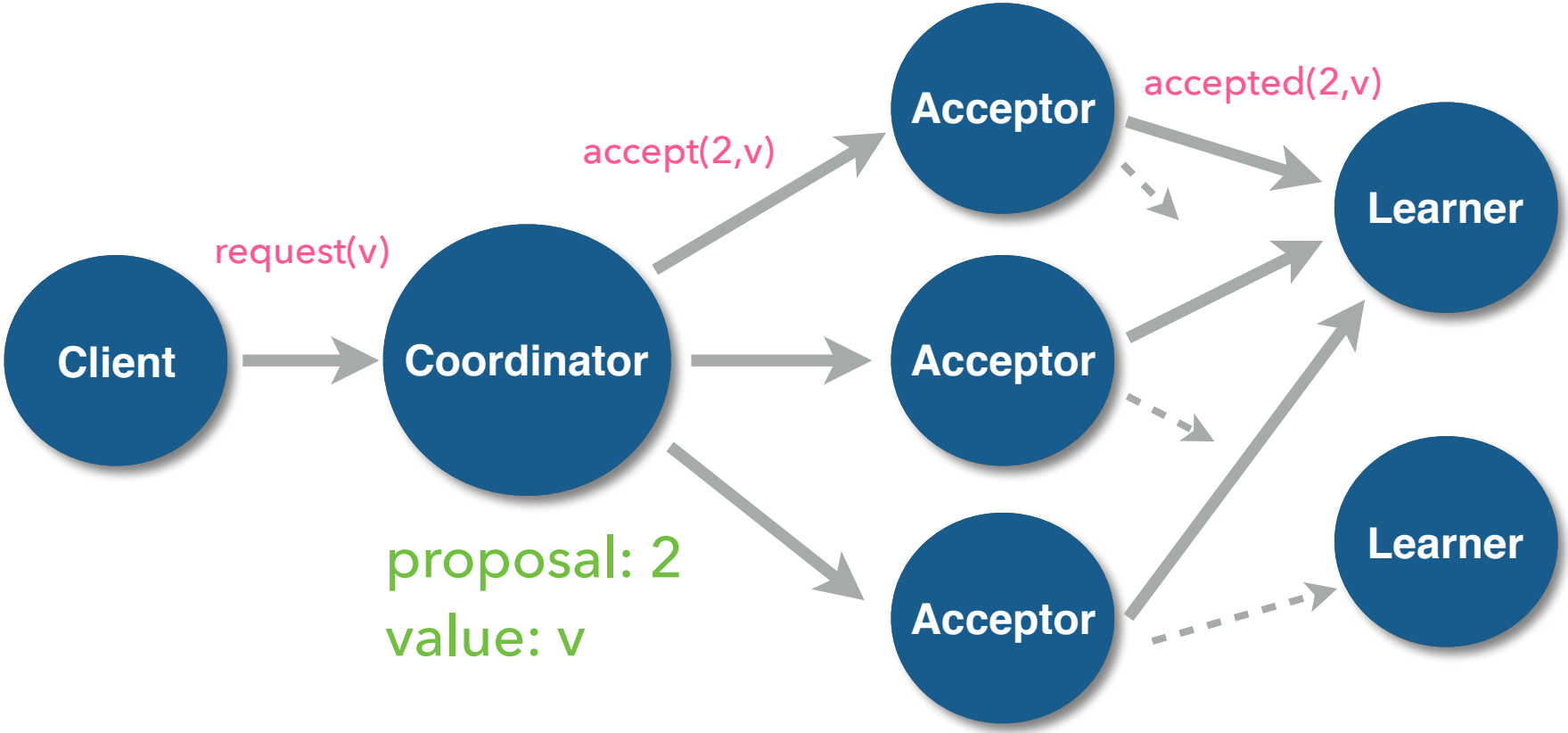
Background on Paxos Consensus



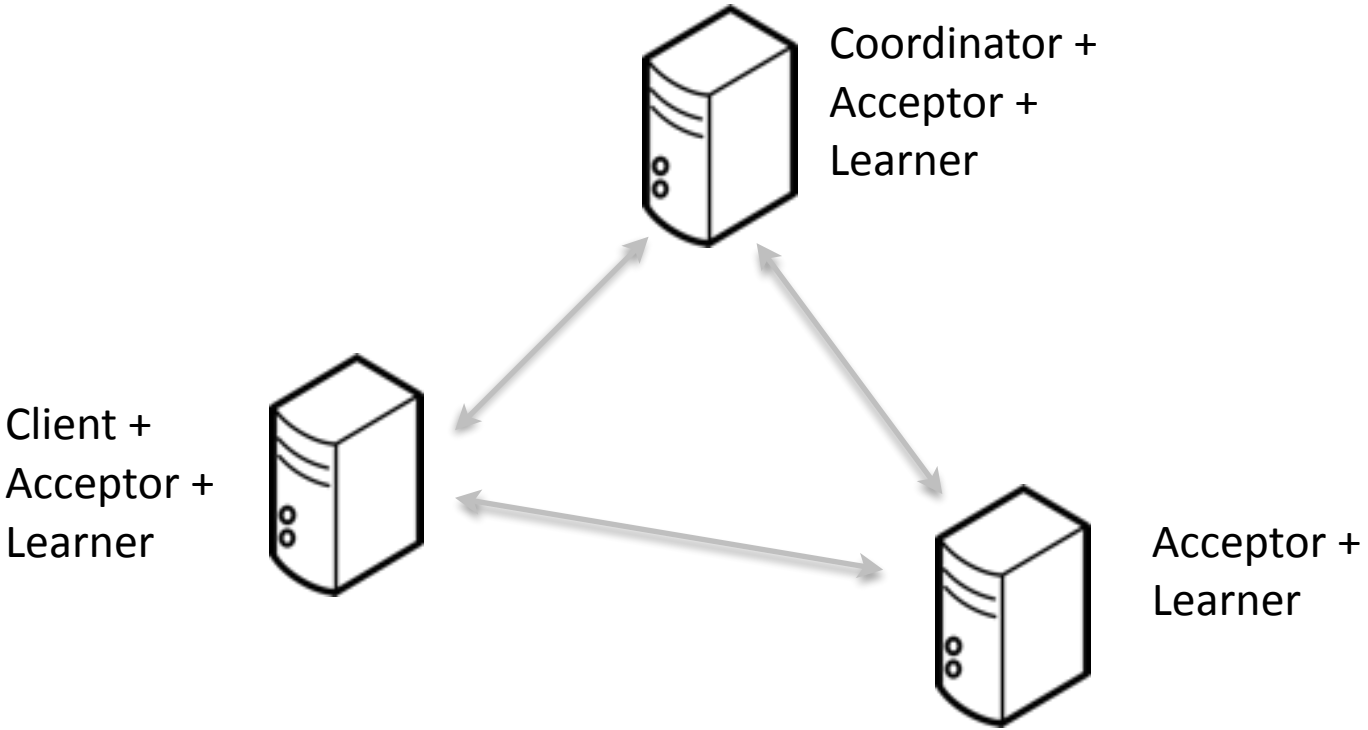
Background on Paxos Consensus



Background on Paxos Consensus



Bottleneck Experiment Setup

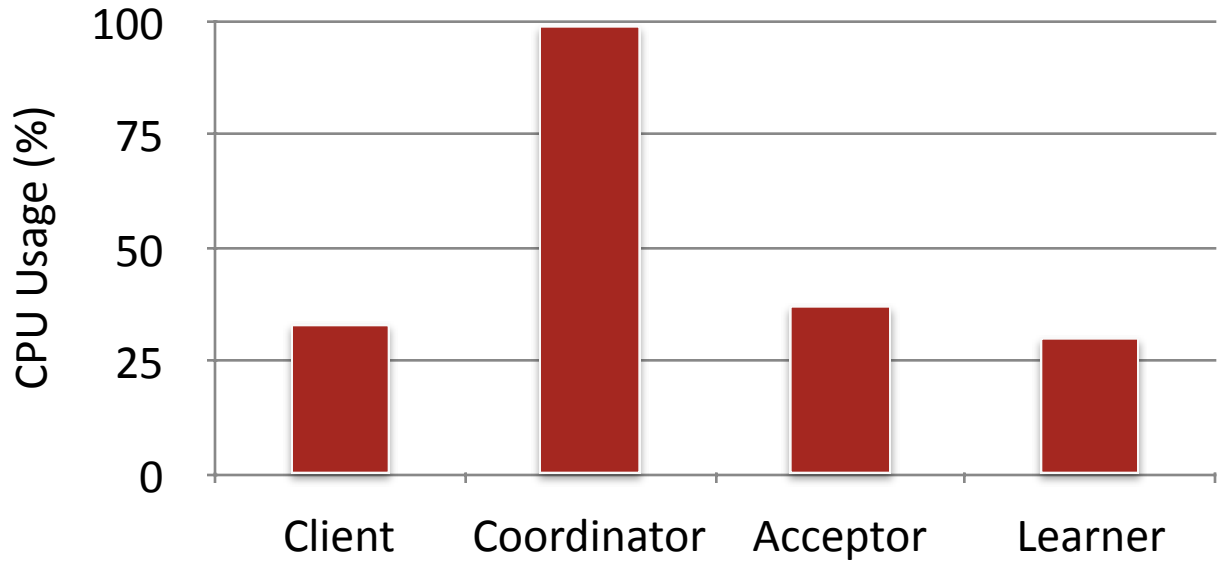


Coordinator Bottleneck

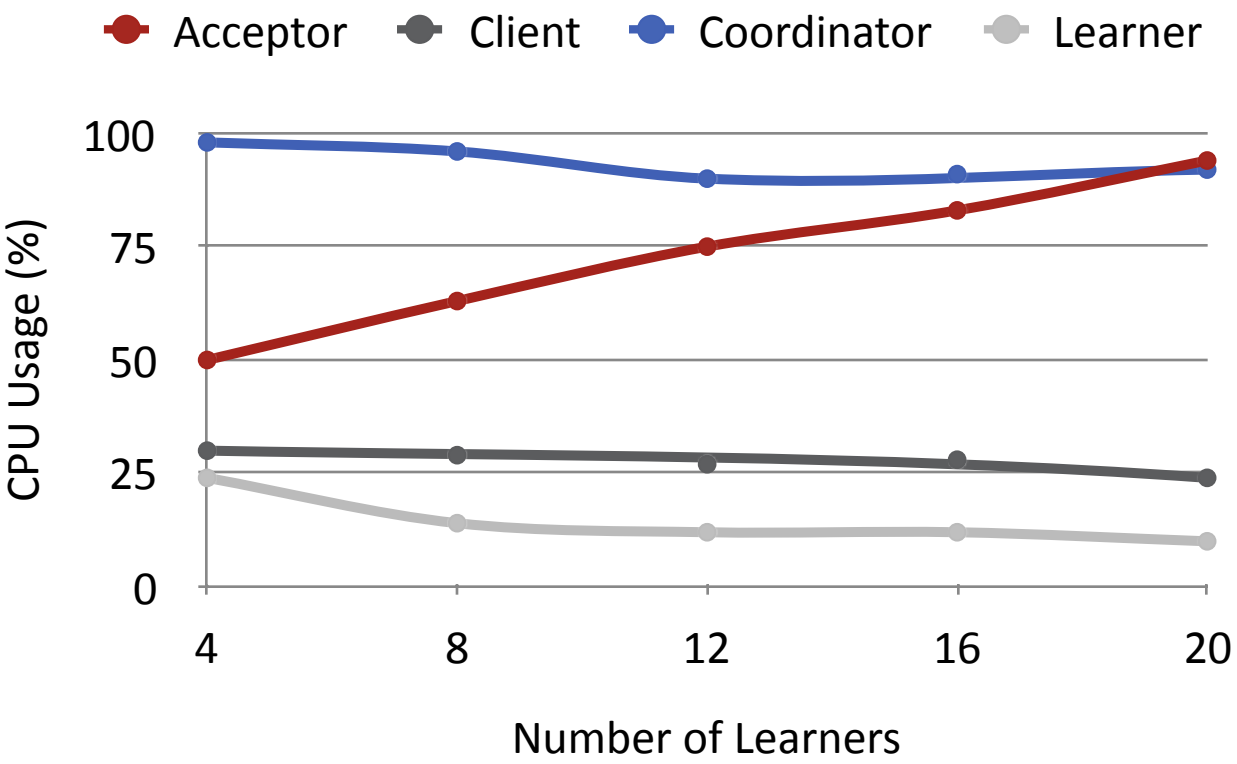
Client offered load at maximum rate

Messages are 102 Bytes

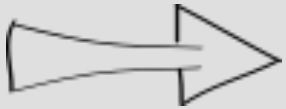
Minimum latency is **96 µs**



Acceptor Bottleneck

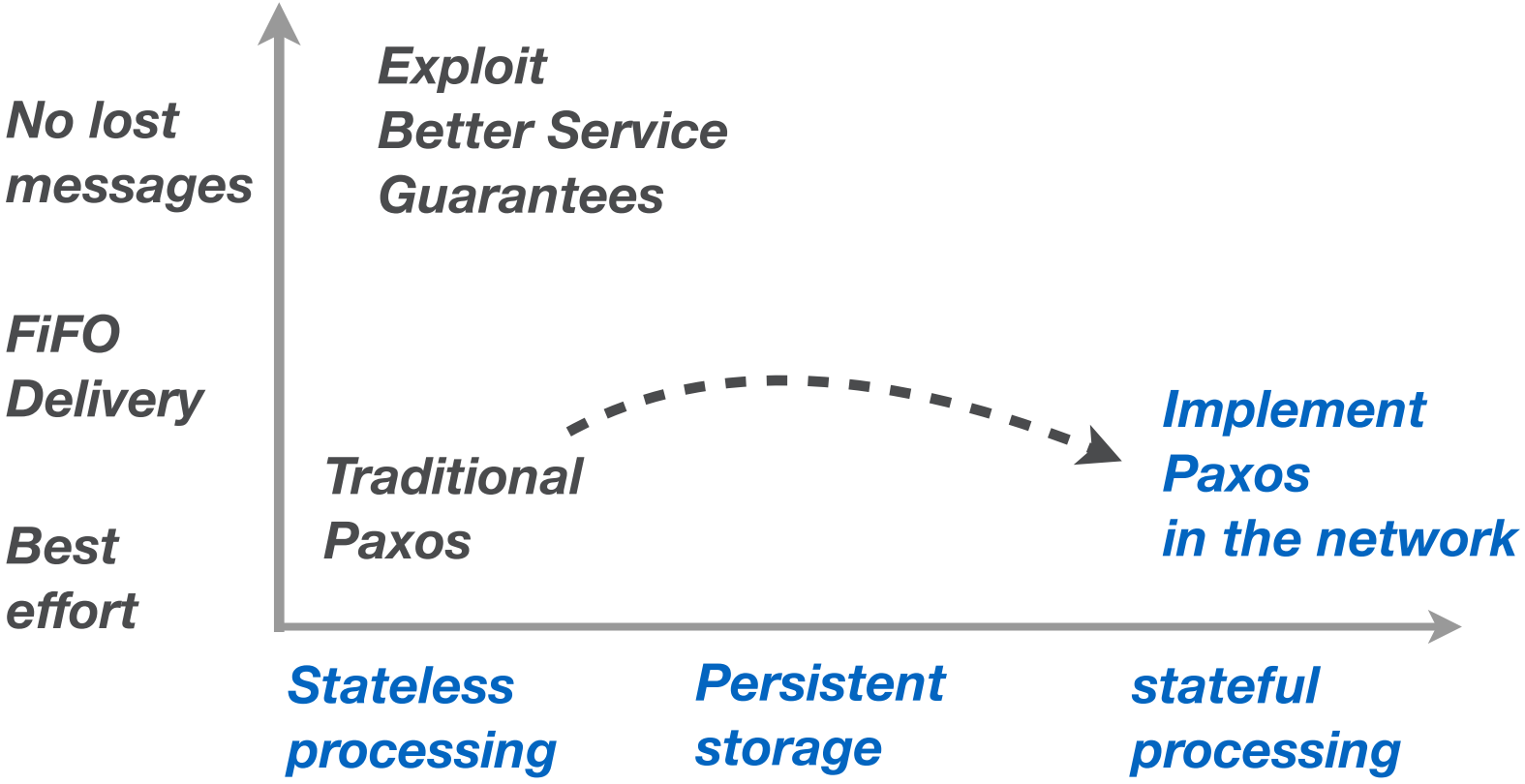


To increase replication factor, we launch multiple learner processes on each servers



CAANS:
Consensus As A Network Service

Consensus / Network Design Space

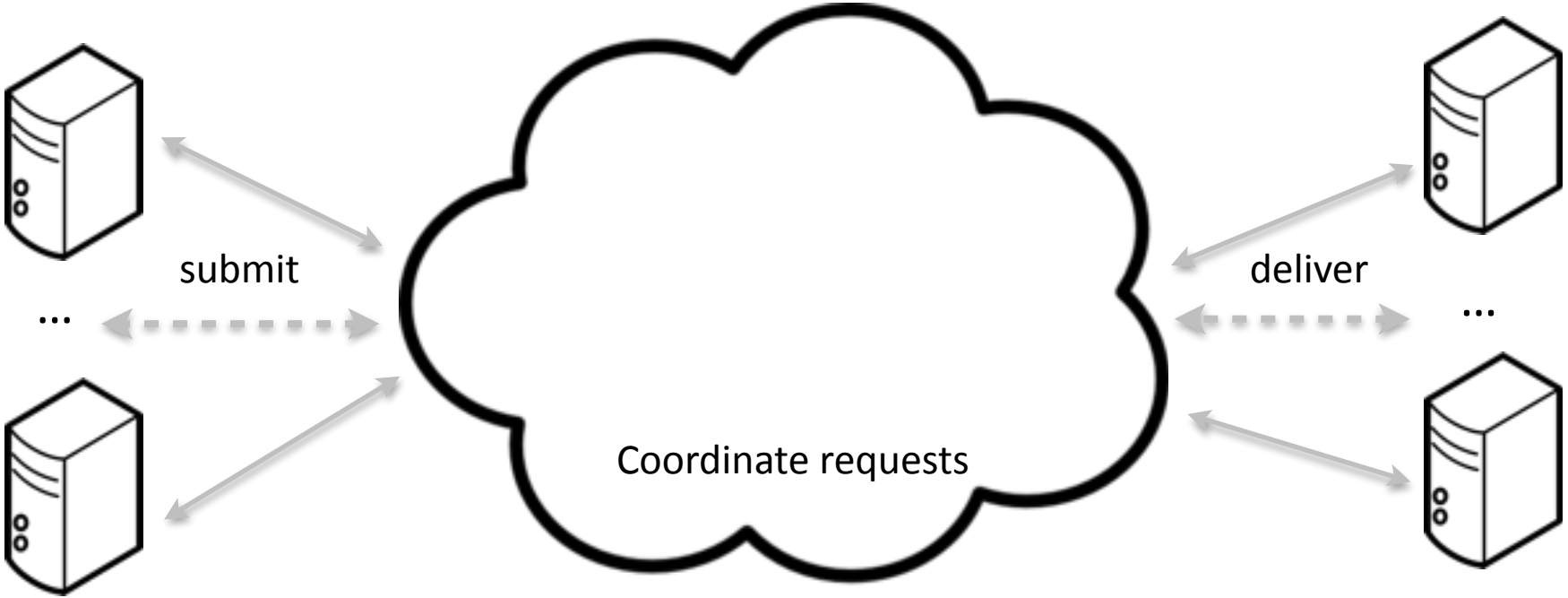


Design Goals

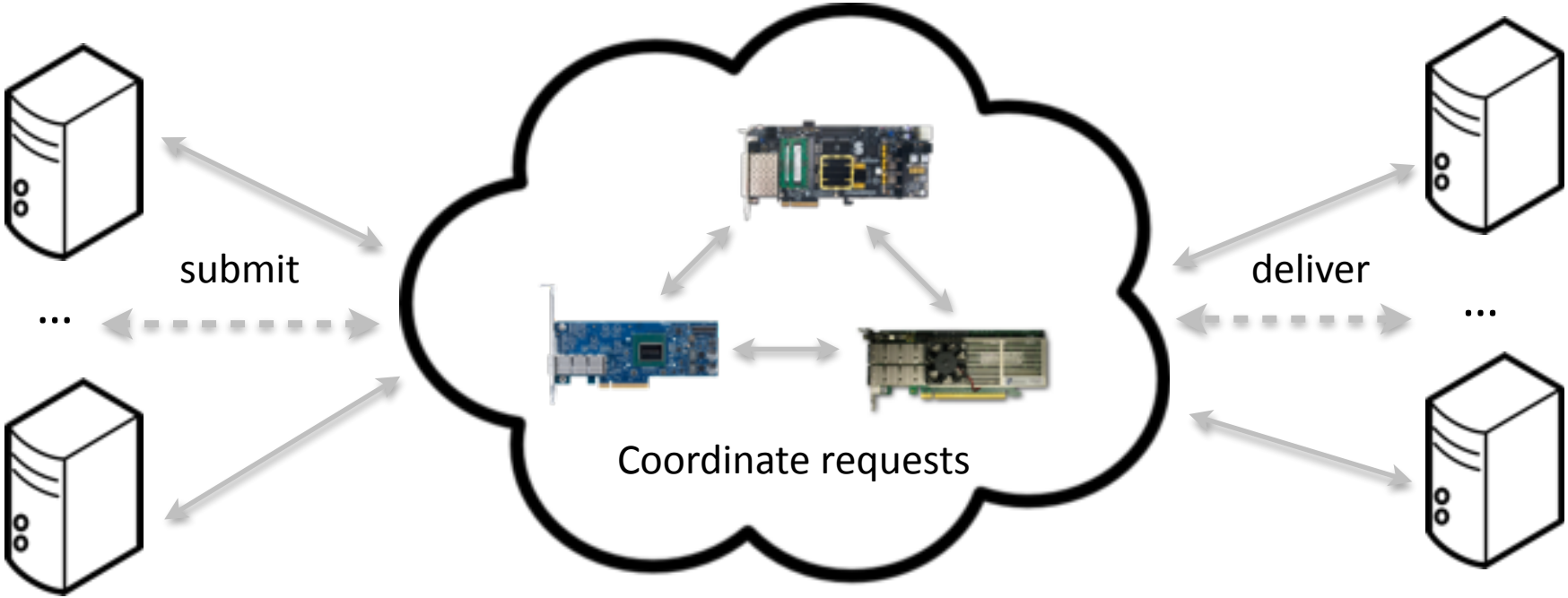
A network consensus library:

- Compatible with the software library
- Independent targets
- High consensus throughput
- Low end-to-end latency

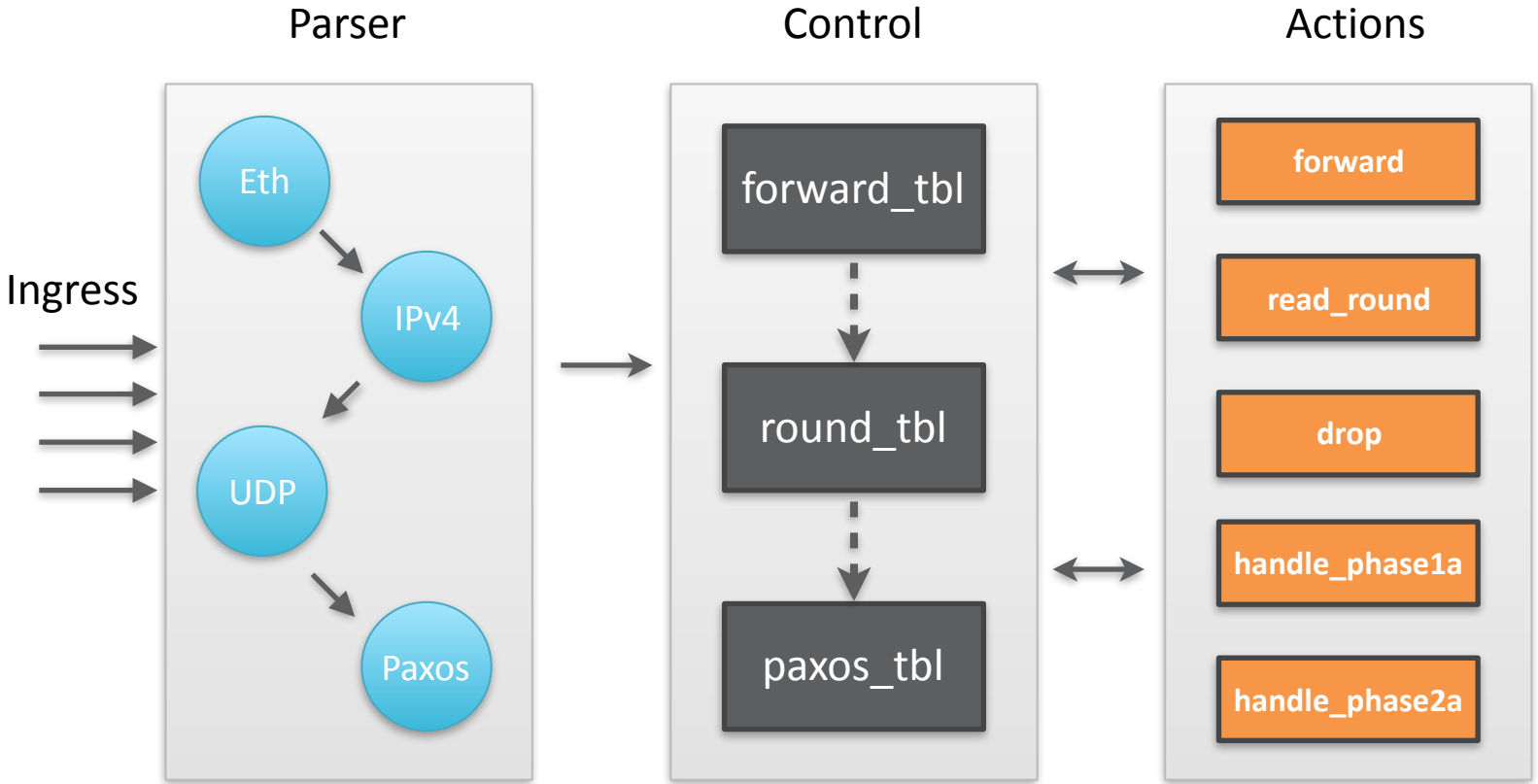
Consensus as a network service



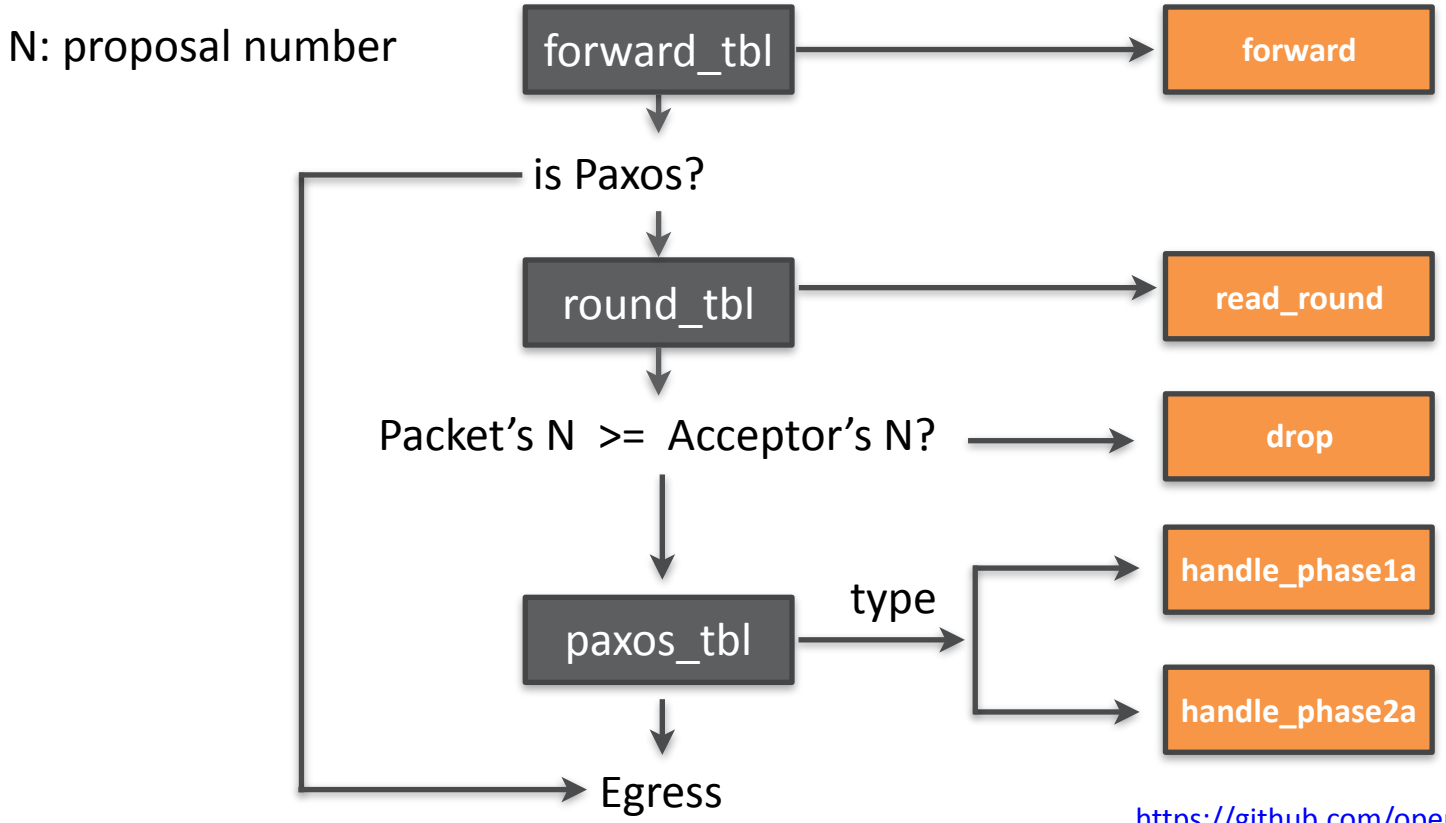
Consensus as a network service



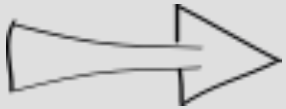
P4Paxos Data Flow



P4Paxos Data Flow



https://github.com/open-nfpsw/NetPaxos

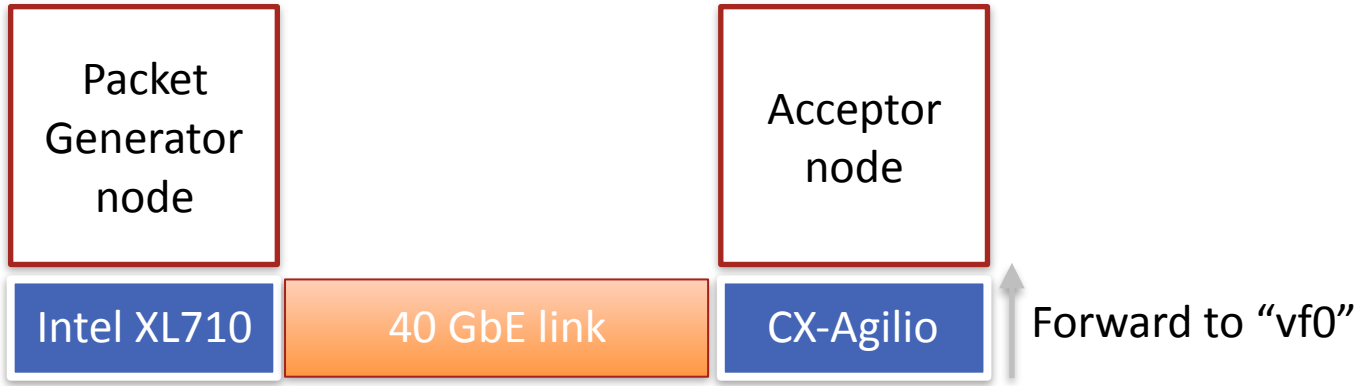


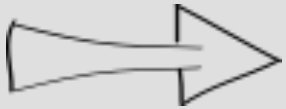
DEMO: Debugging Paxos Acceptor

DEMO

Paxos Acceptor

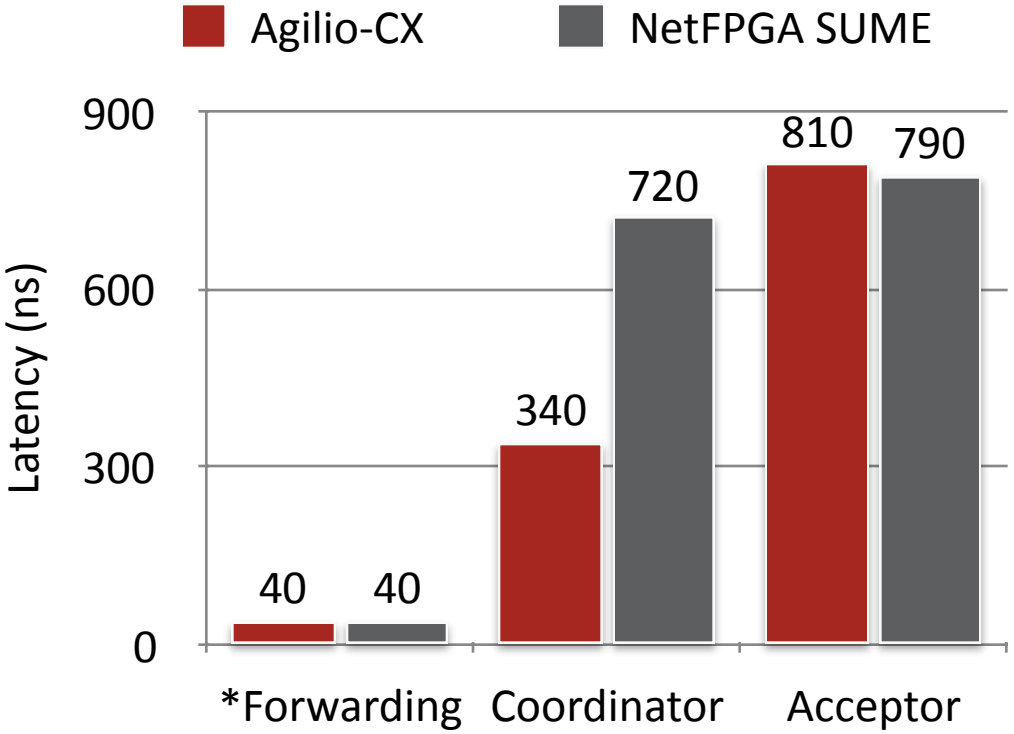
- Store Accept Message in Registers
- Modify Paxos header
- Read accepted proposal from Registers



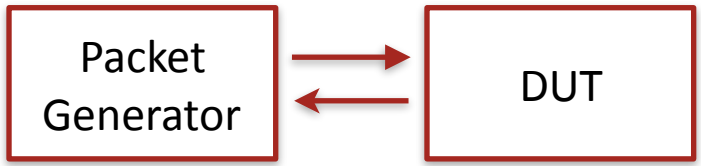


Results: Absolute Performance

Result: Forwarding Latency



*Numbers are provided by other sources



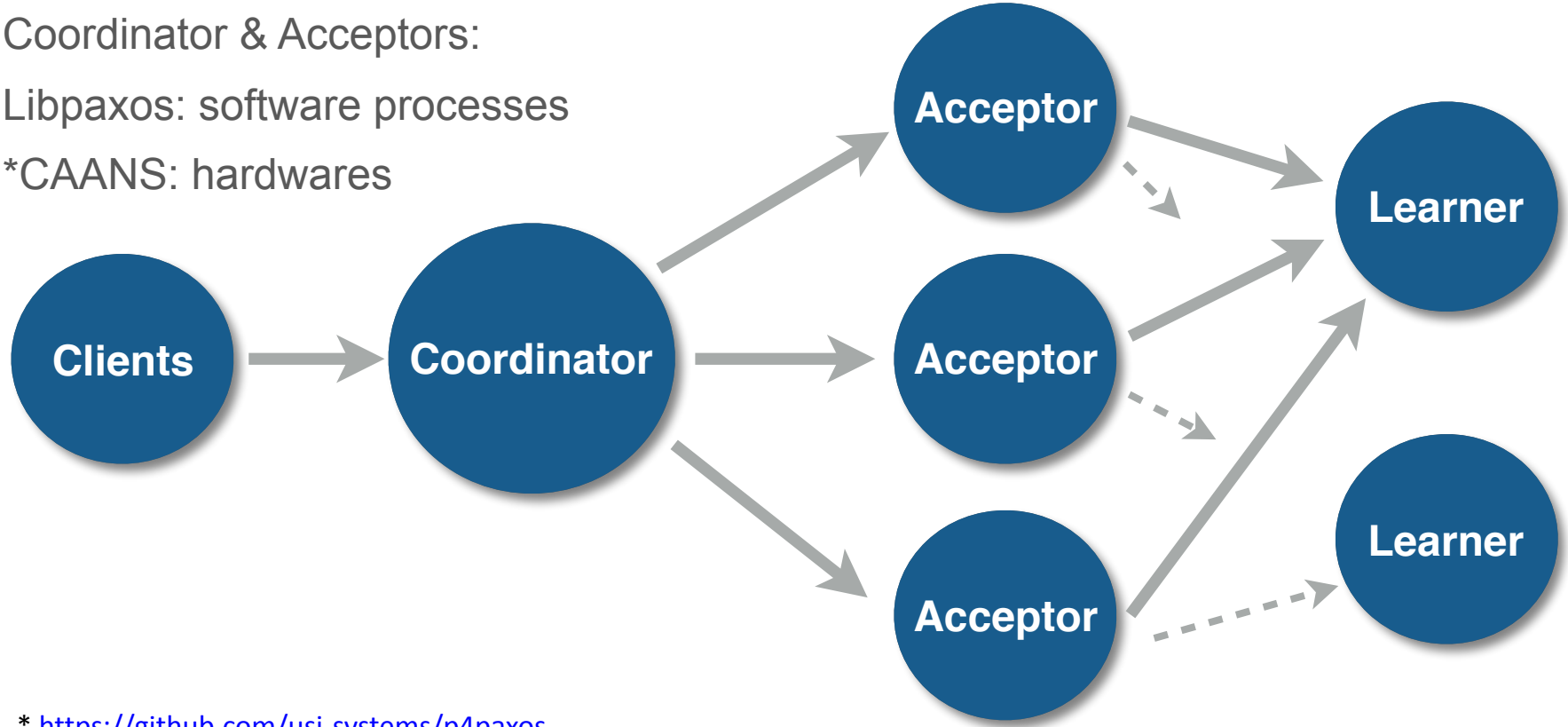
software 96µs
vs.
hardware 340ns

Experiment Setup

Coordinator & Acceptors:

Libpaxos: software processes

*CAANS: hardwares

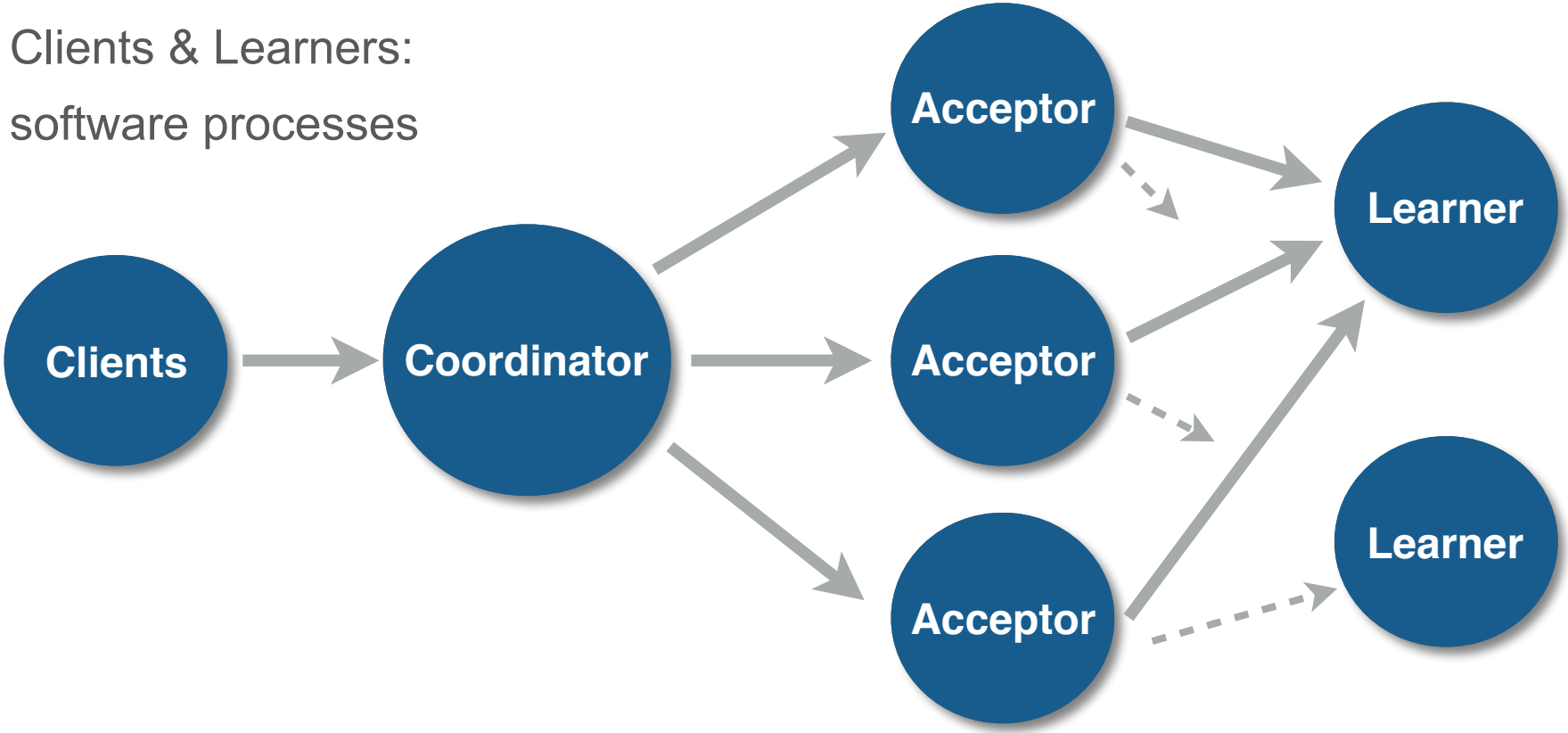


* https://github.com/usi-systems/p4paxos

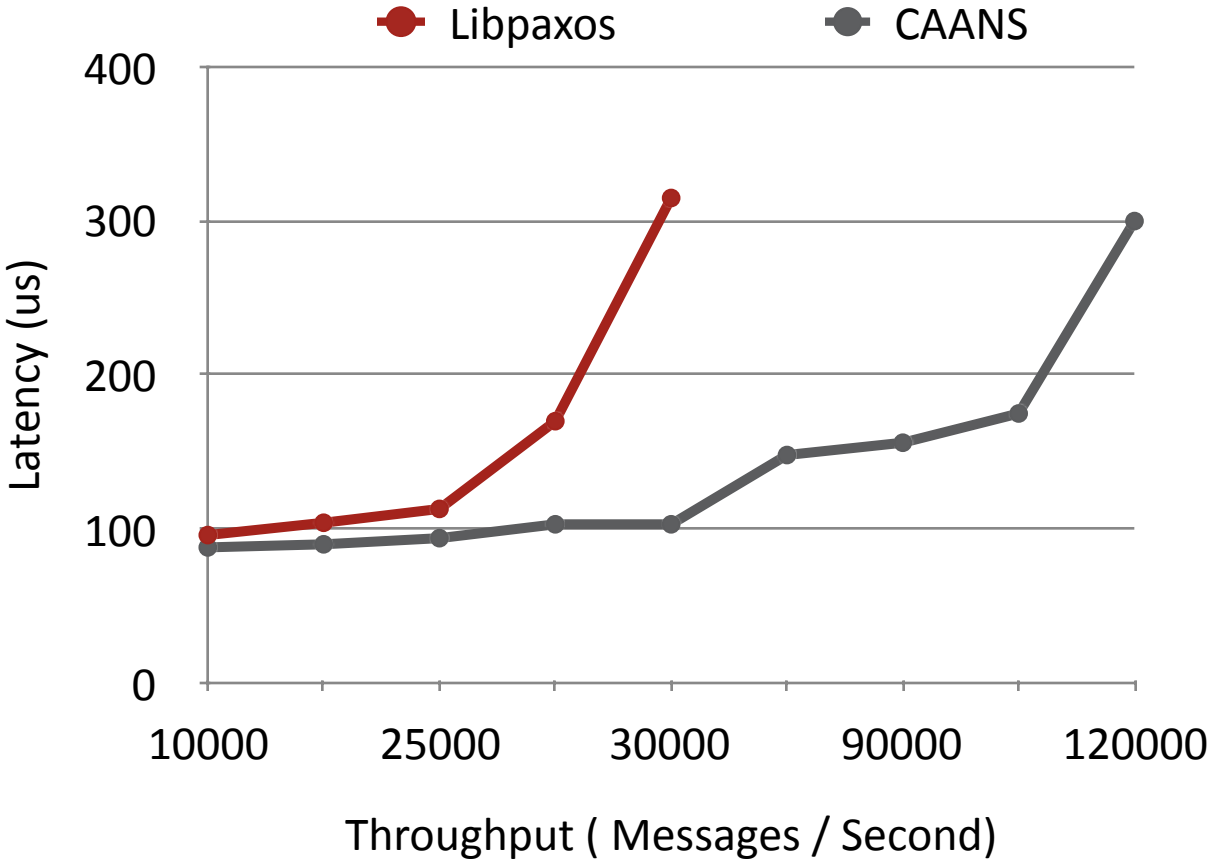
Experiment Setup

Clients & Learners:

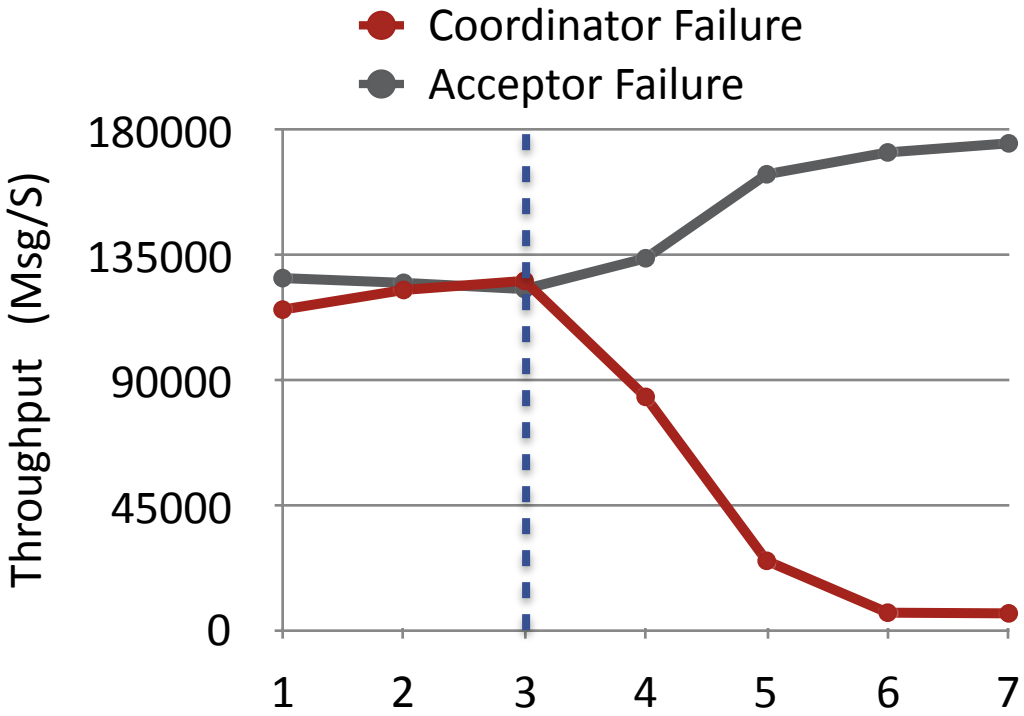
software processes



End-to-End Latency



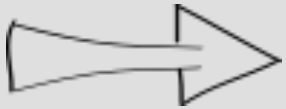
CAANS: Fault Tolerance



Links are down at the 3rd second.

Coordinator failure: requests are switched to a software coordinator

Acceptor failure: Paxos tolerates failure of a minority of nodes



Summary

Summary

Consensus is important in distributed systems

- Optimizations exploit network assumptions

SDN allows network programmability

- OpenFlow, P4

CAANS: Co-design Network and Consensus

- Implement Paxos in network devices
- Achieve high performance

Future Work

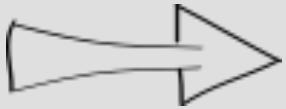
- Checkpoint Acceptor’s log
- Fast fail-over to software/hardware coordinator
- Use kernel-bypass API to increase learner’s performance
- Full-fledged deployment in traditional networks

Acknowledgements

- Thank Bapi Vinnakota, Mary Pham, Jici Gao, and Netronome for providing us with a hardware testbed, and support with using their toolchain

- Thank Gordon Brebner and Xilinx for donating two SUME boards, providing access to SDNet, performing measurements

- Thank Google! Faculty award helped support this research



The End

Huynh Tu Dang,
Università della Svizzera italiana (USI)

tudang.github.io

NetPaxos

http://www.inf.usi.ch/faculty/soule/netpaxos.html