Western Digital®

Achieving Fault Tolerance for Persistent Memory

Huynh Tu Dang Jaco Hofmann, Yang Liu, Marjan Radi, Dejan Vucinic, Fernando Pedone, and Robert Soulé

Principal Research Engineer

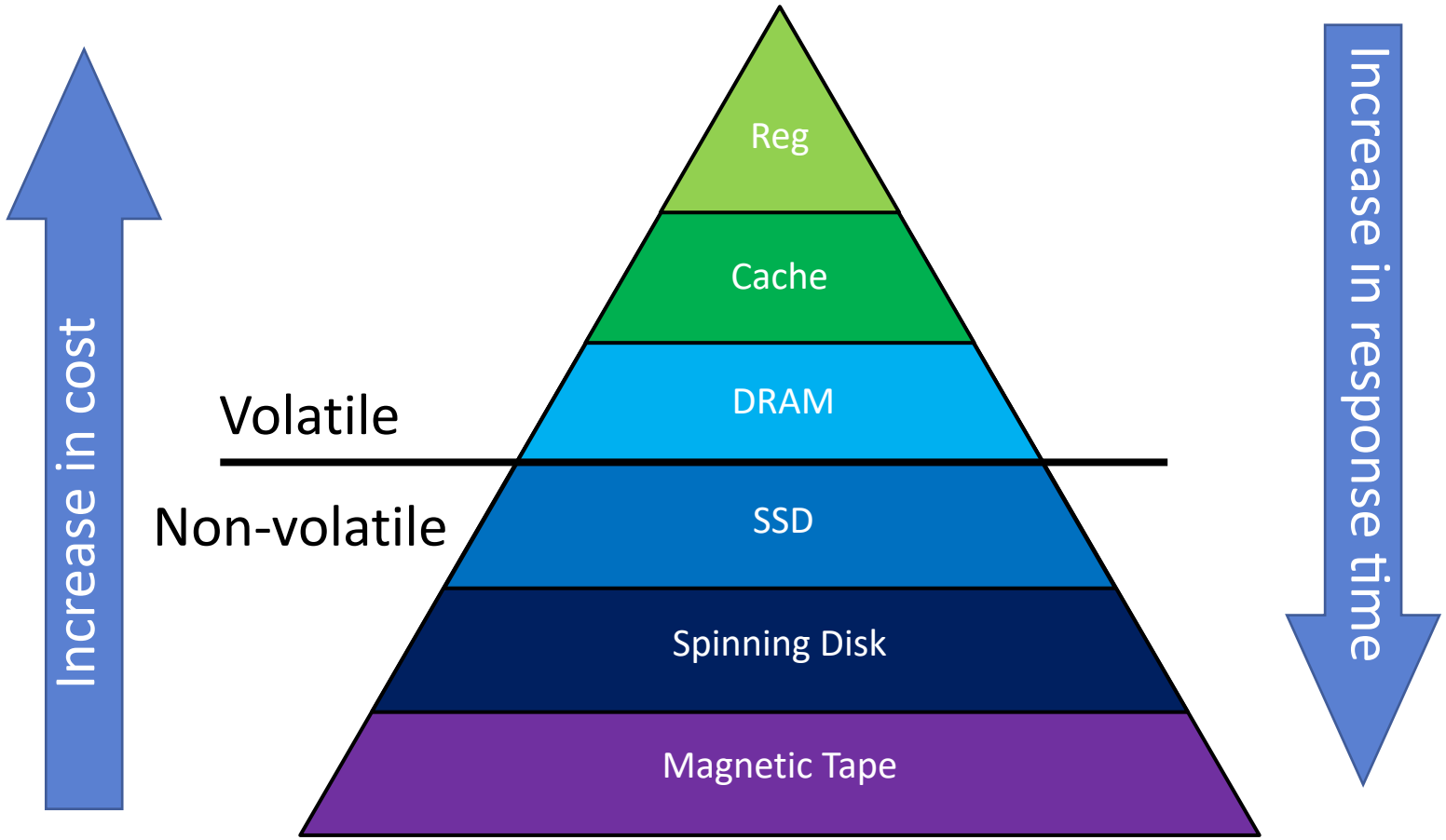
August 6th, 2019

Agenda

1. Memory Hierarchy & Persistent Memory
2. Protocol Implementing Atomic Register by Attiya, Bar-Noy, Dolev
3. PISA: Protocol Independent Switch Architecture
4. System Design & Evaluation
5. Conclusions

Memory Hierarchy

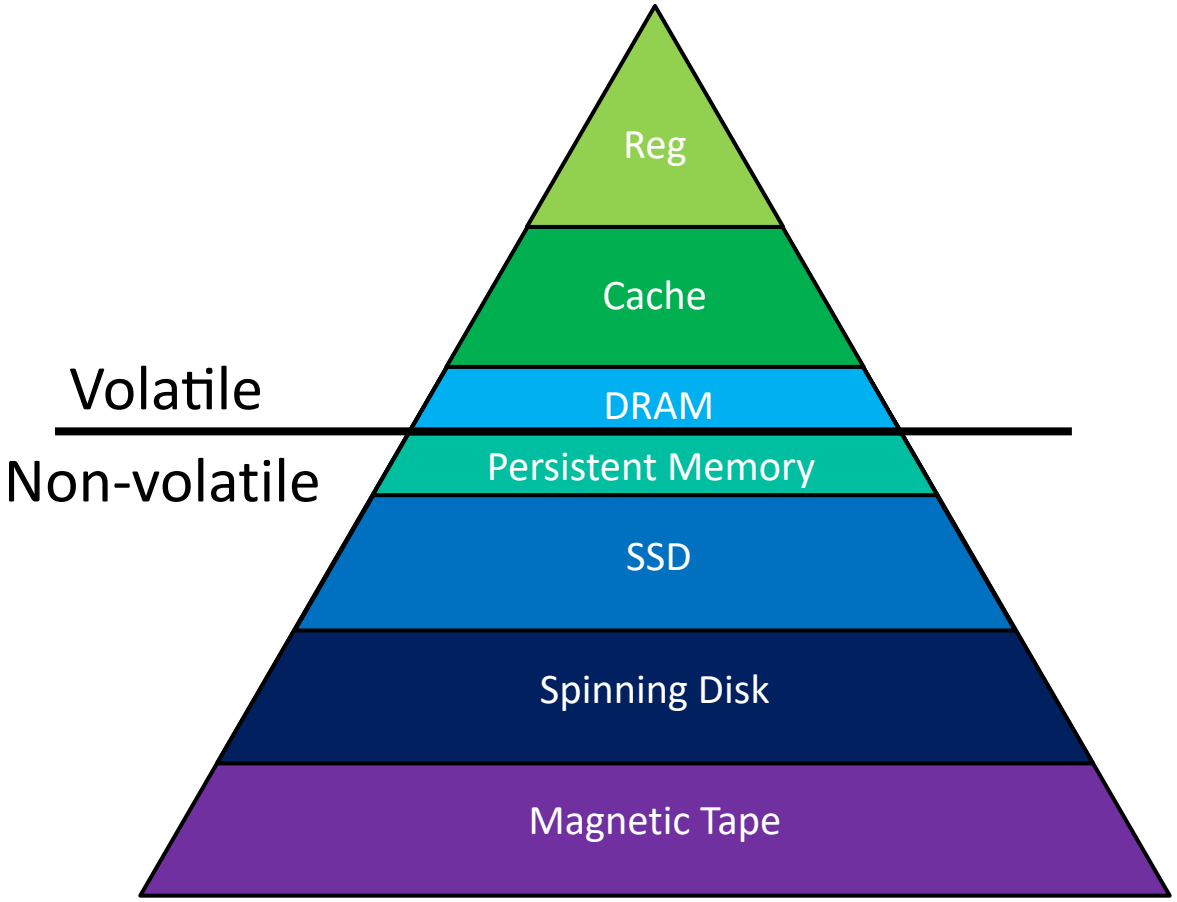
- Volatile memories
  - ✚ Low response time
  - ➖ High cost
  - ➖ High power consumption
- Non-volatile memories
  - ✚ Low cost
  - ✚ Low power consumption
  - ➖ High response time



Persistent Memory (PM)

Phase-Change Memory (PCM), Resistive RAM (ReRAM), and Spin-Torque Magnetic RAM (STT-MRAM)

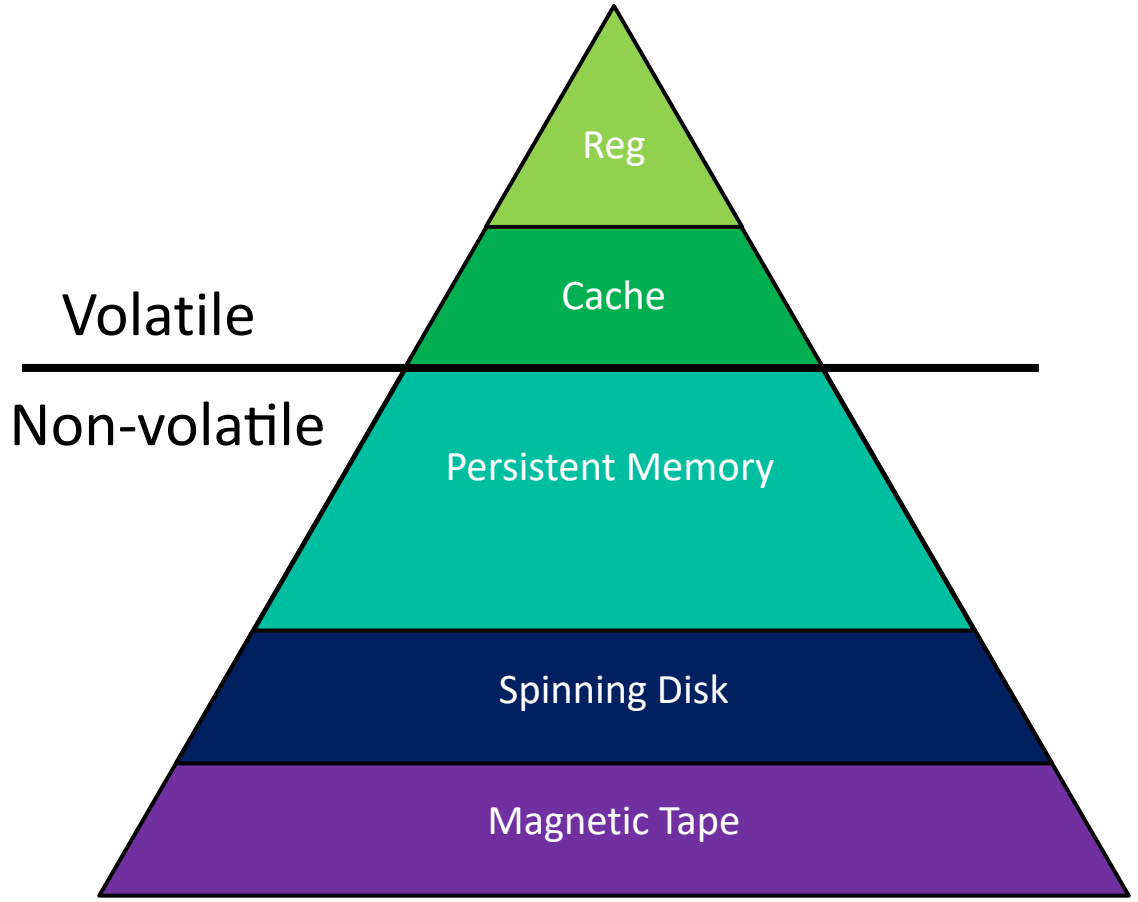
- Non-volatile
- Byte-addressable
- DRAM-like response time
- Cost significant-less than DRAM



Pros and Cons of Persistent Memory

Persistent memory could replace several tiers of the tradition memory/storage

- Pros:
  - Non-volatile memory simplifies architecture and algorithm design
  - Cost advantage over DRAM reduces CAPEX of data centers
- Cons:
  - Persistent memory technologies are subject to wear-out mechanisms
  - Persistent memory imposes practical limits on scaling out

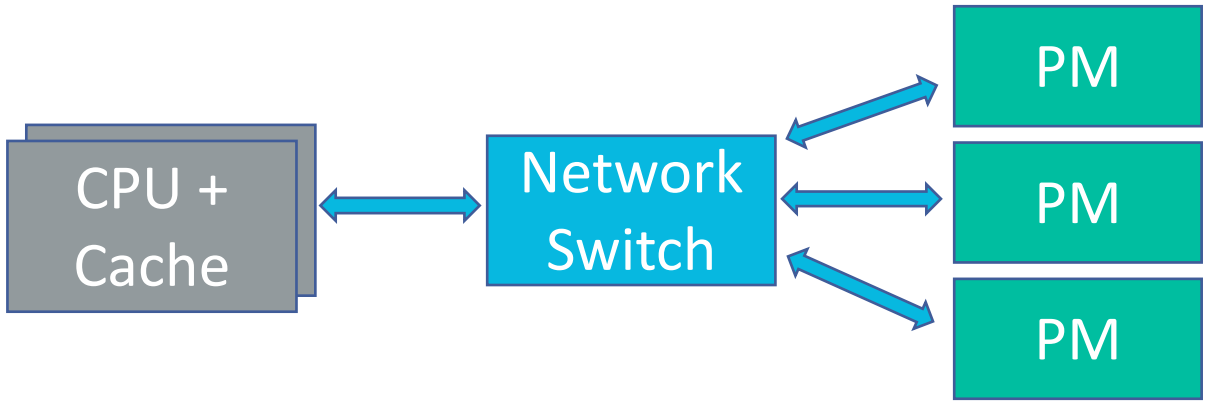


Mechanisms for Handling Memory Faults

Storage Medium	Approach	Problem
CPU main memory	Ignore failure	System crashes
Super computer main memory	Checkpointing	Complicated management and cost
Disk and SSD	RAID	Centralized controller doesn't scale well

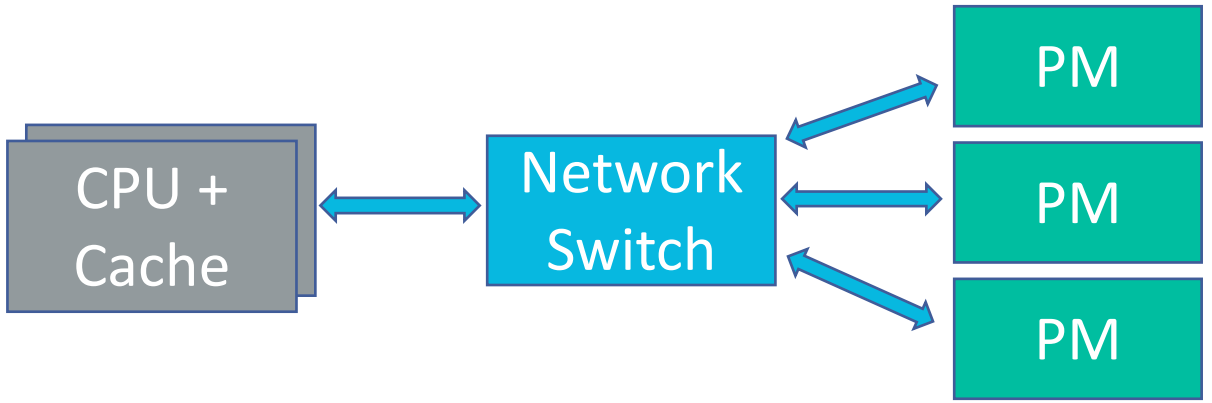
New Approach to Provide Fault Tolerance for Non-volatile Main Memory

- Treat memory as a distributed system
  - Replicate data to cope with failures
  - Use consensus protocol for consistency
- Great promise for performance
  - E.g., NetPaxos, NetChain, Speculative Paxos, Consensus in a Box demonstrate tremendous reductions in latency



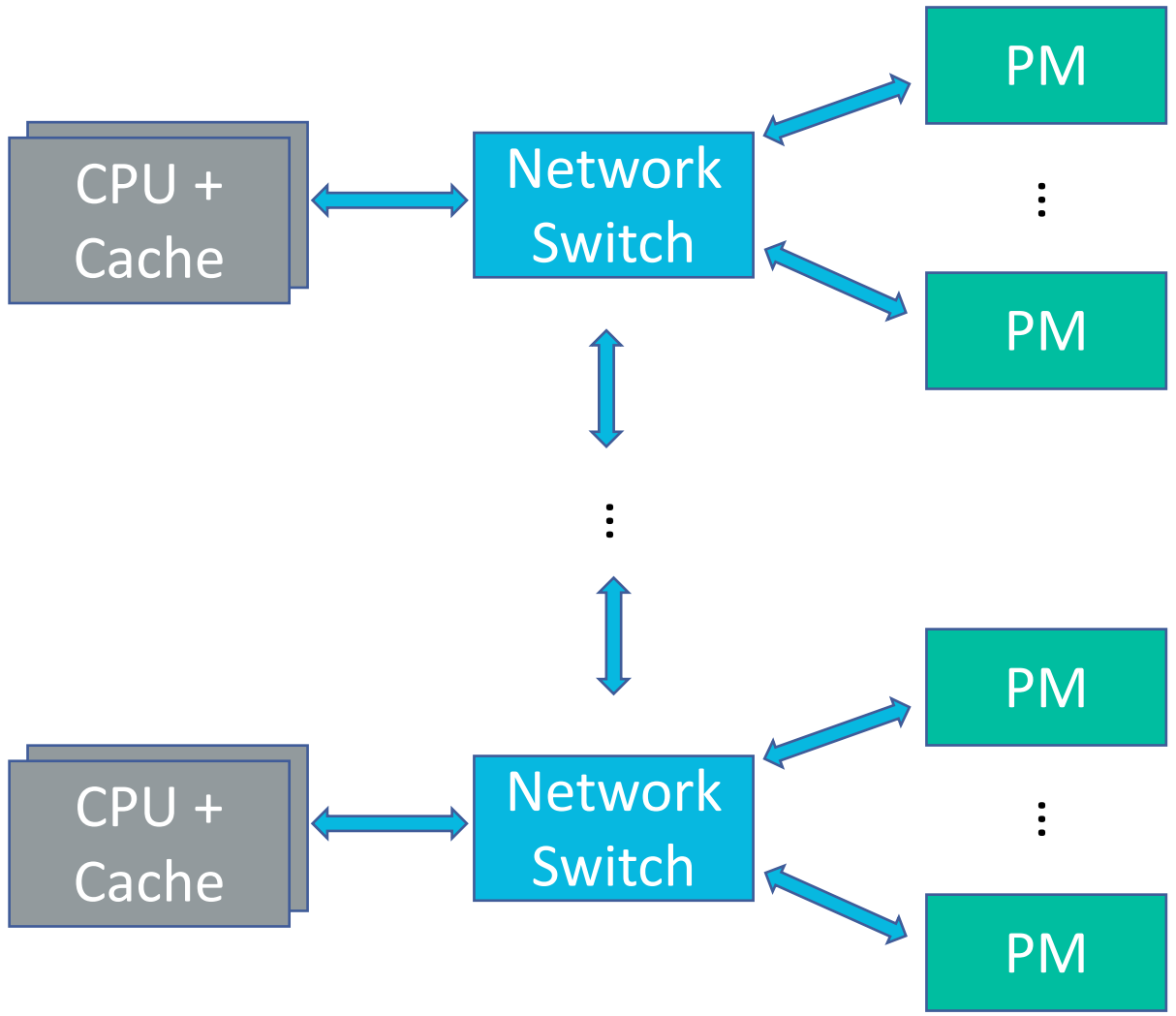
New Approach to Provide Fault Tolerance for Non-volatile Main Memory

- Use a generalization of a protocol by Attiya, Bar-Noy and Dolev (ABD)
- Well-suited to the task for 3 reasons:
  - Simple protocol: supports only Read and Write operations
  - Straightforward for an efficient in-network implementation
  - Less state: stores replicated data and a logical timestamp



Long-Term Goals

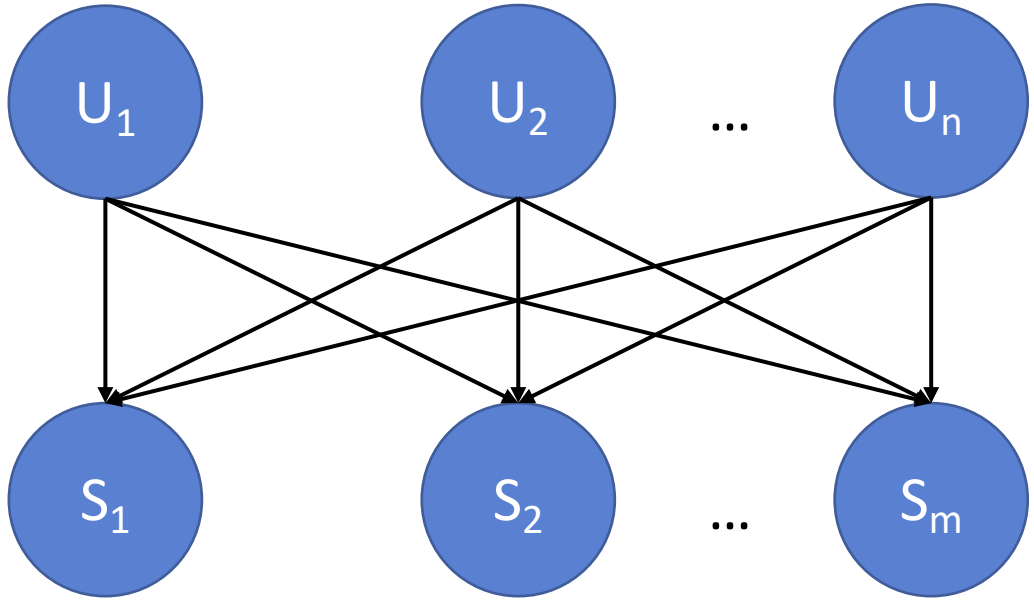
- Build a scalable non-volatile main memory system
- Offer Zettabyte memory capacity
- Tolerate arbitrary CPU, network and memory failures



Protocol Implementing Atomic Register by Attiya, Bar-Noy, Dolev

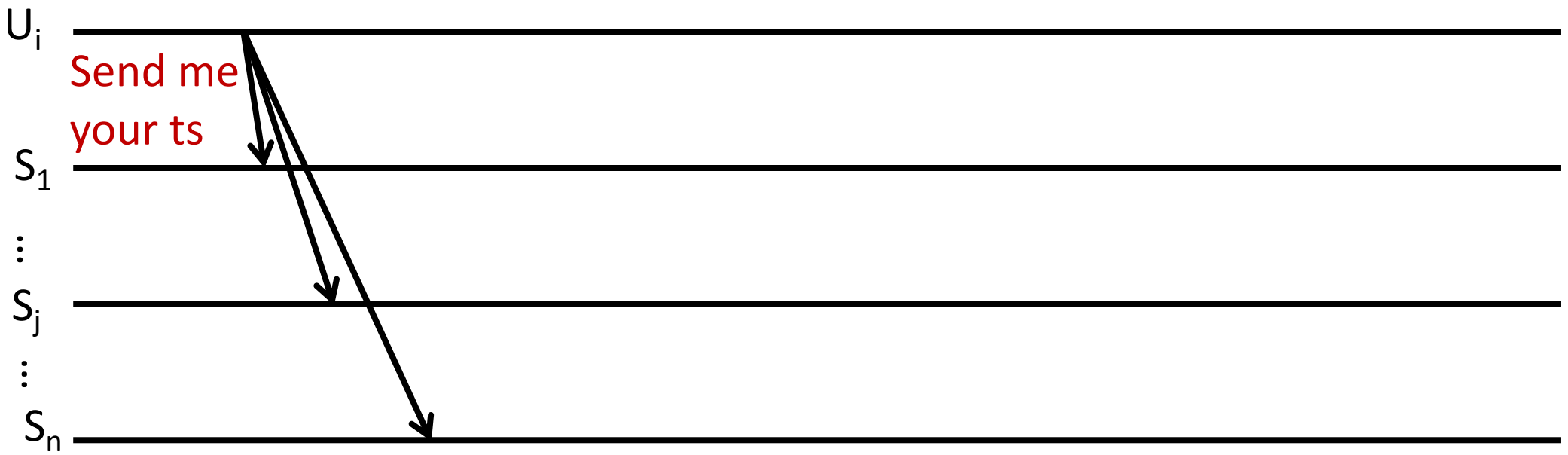
Attiya, Bar-Noy, Dolev (aka. ABD) protocol

- Implement an atomic register in an asynchronous system
- Is more efficient in terms of communication steps than Paxos
- Emulate shared memory with message passing
- Generalized to support for multiple writers and multiple readers



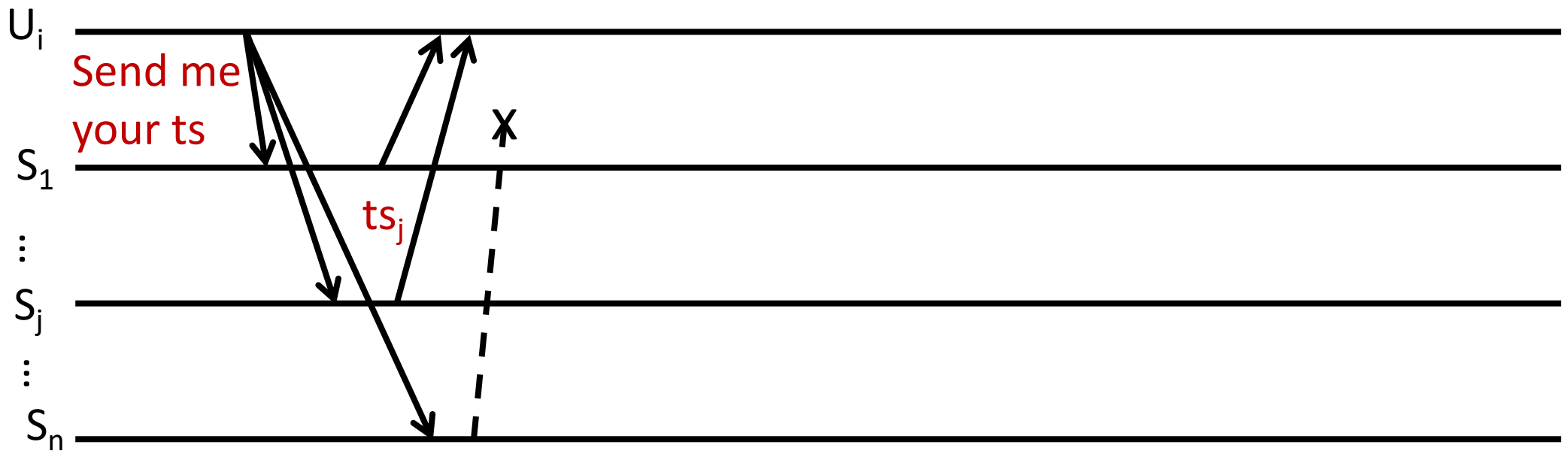
Write Operation

Get TimeStamp from all servers



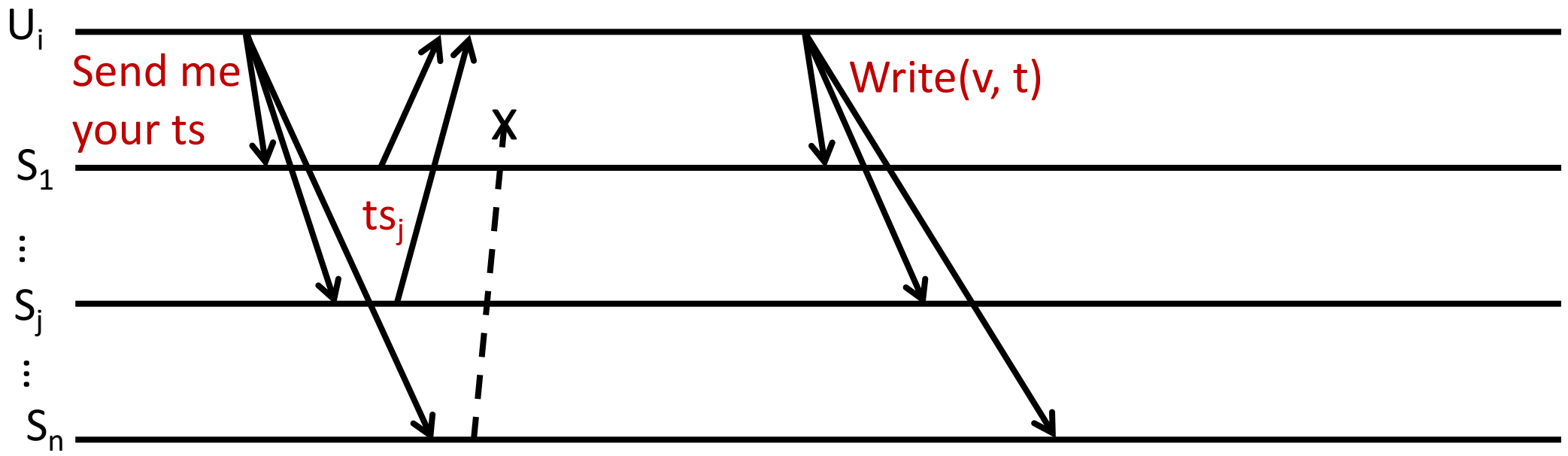
Write Operation

Chooses t such that t is bigger than any previous t and any ts_i



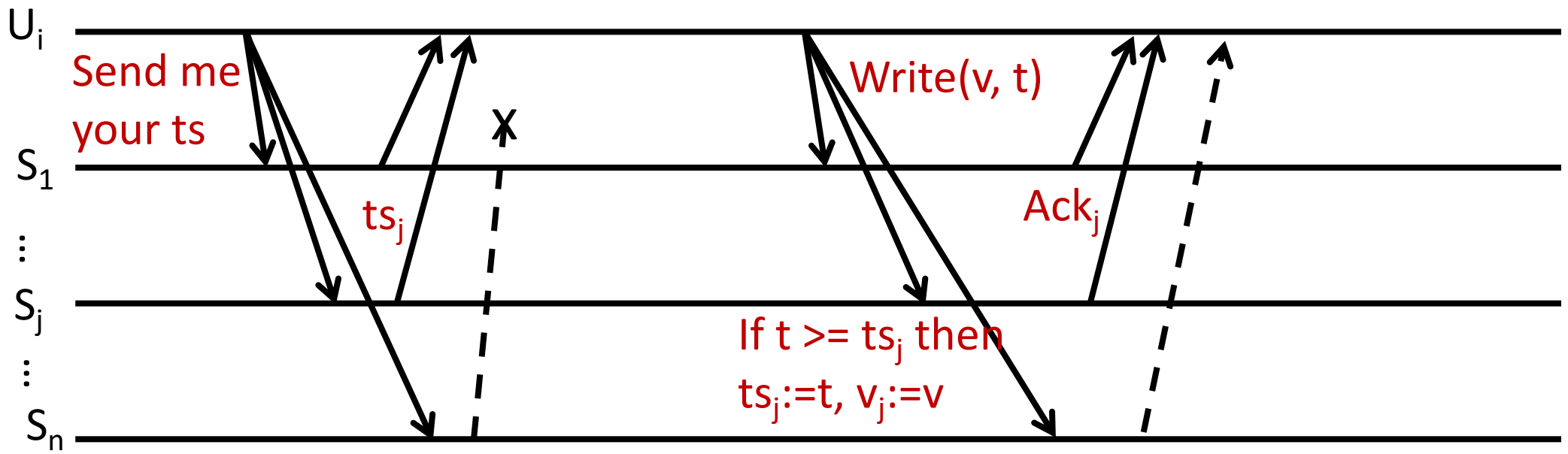
Write Operation

Send Write requests to all servers

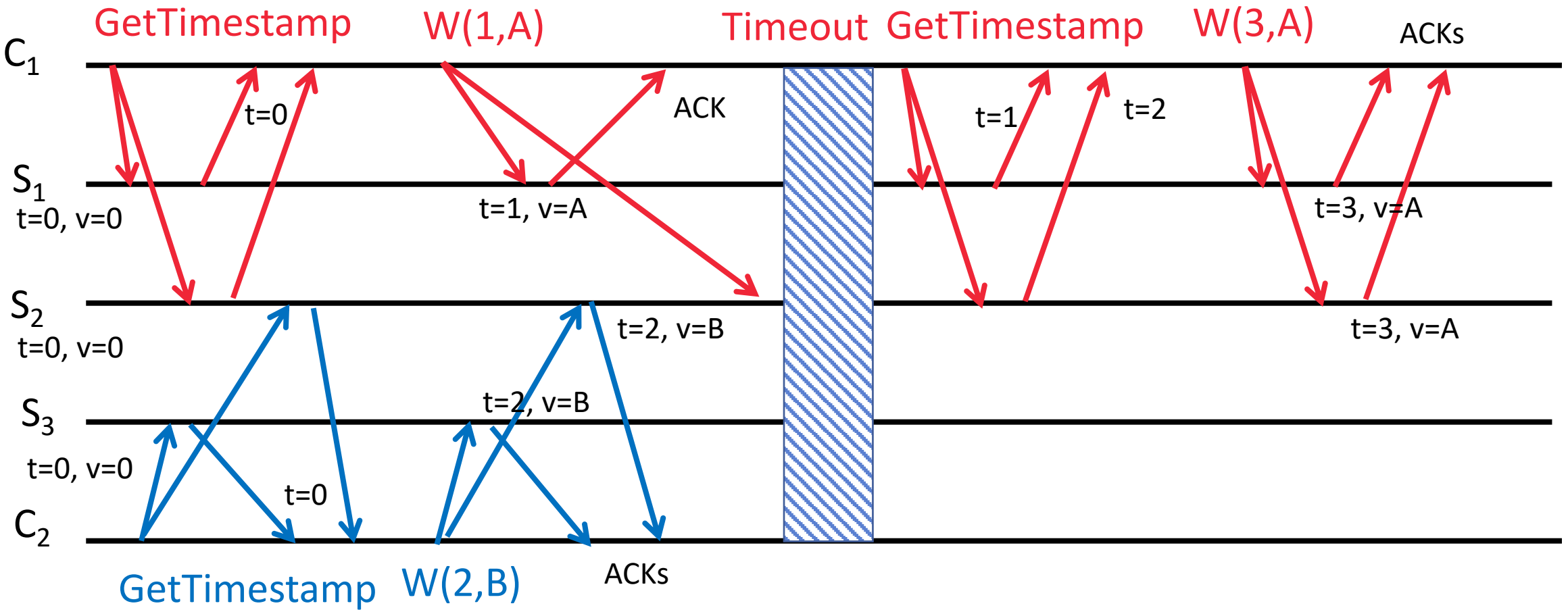


Write Operation

Servers send acknowledgement to the client

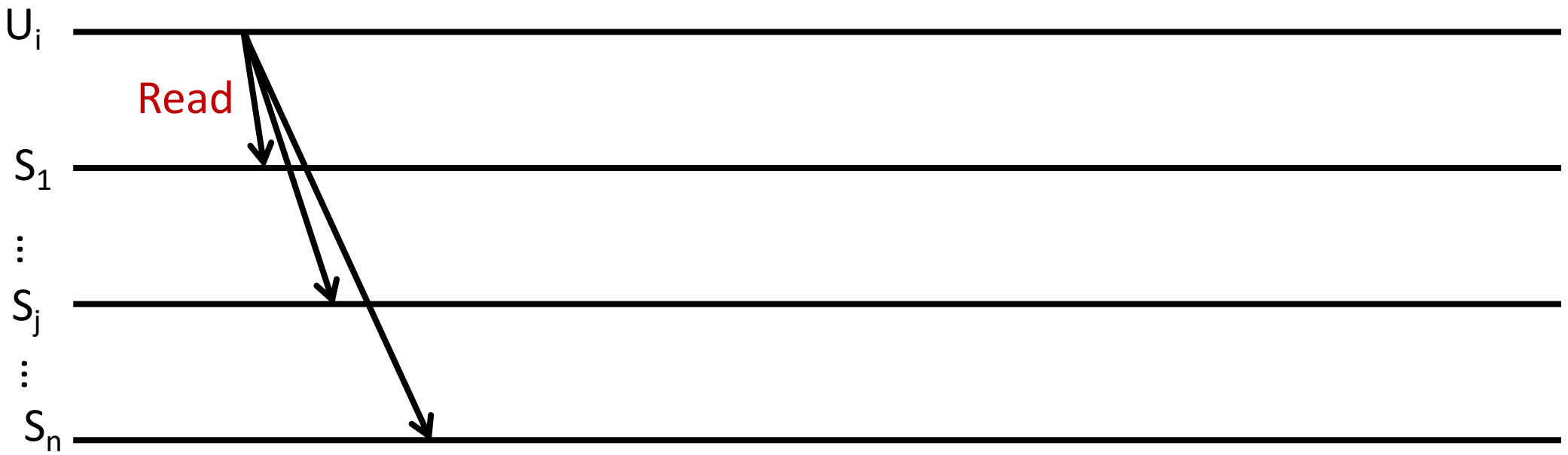


Concurrent Writes



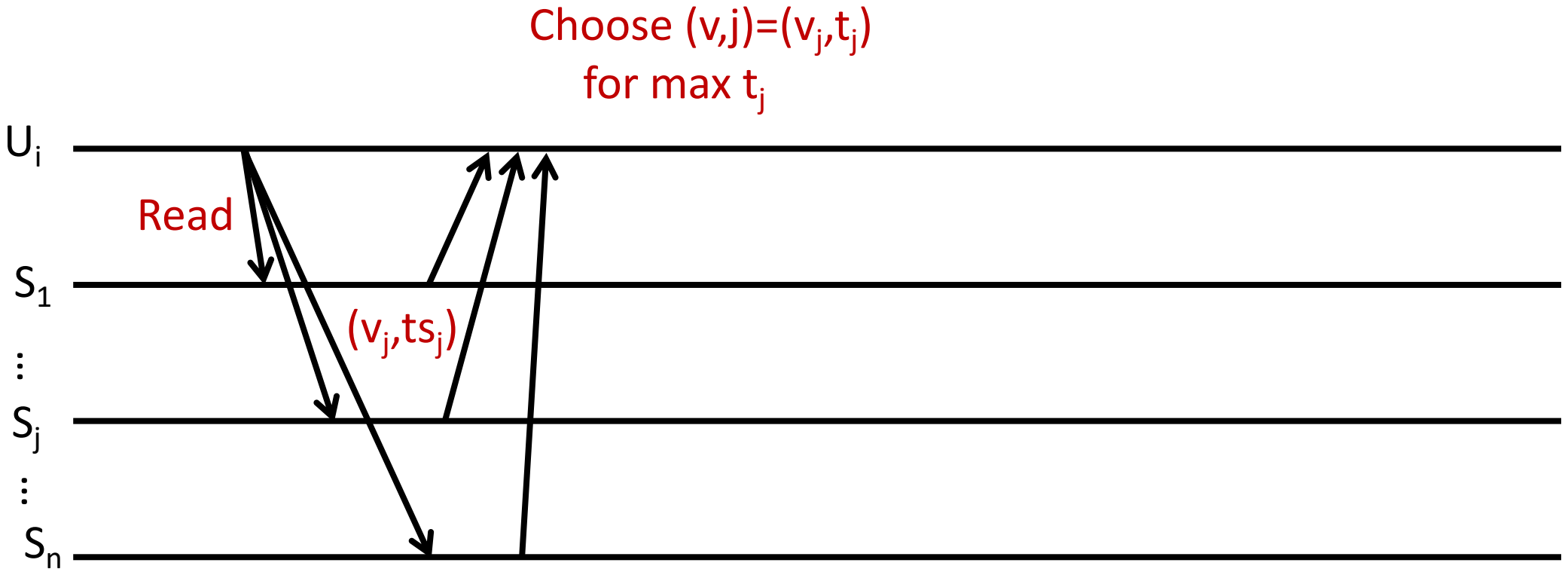
Read Operation

Read value and timestamp from all servers



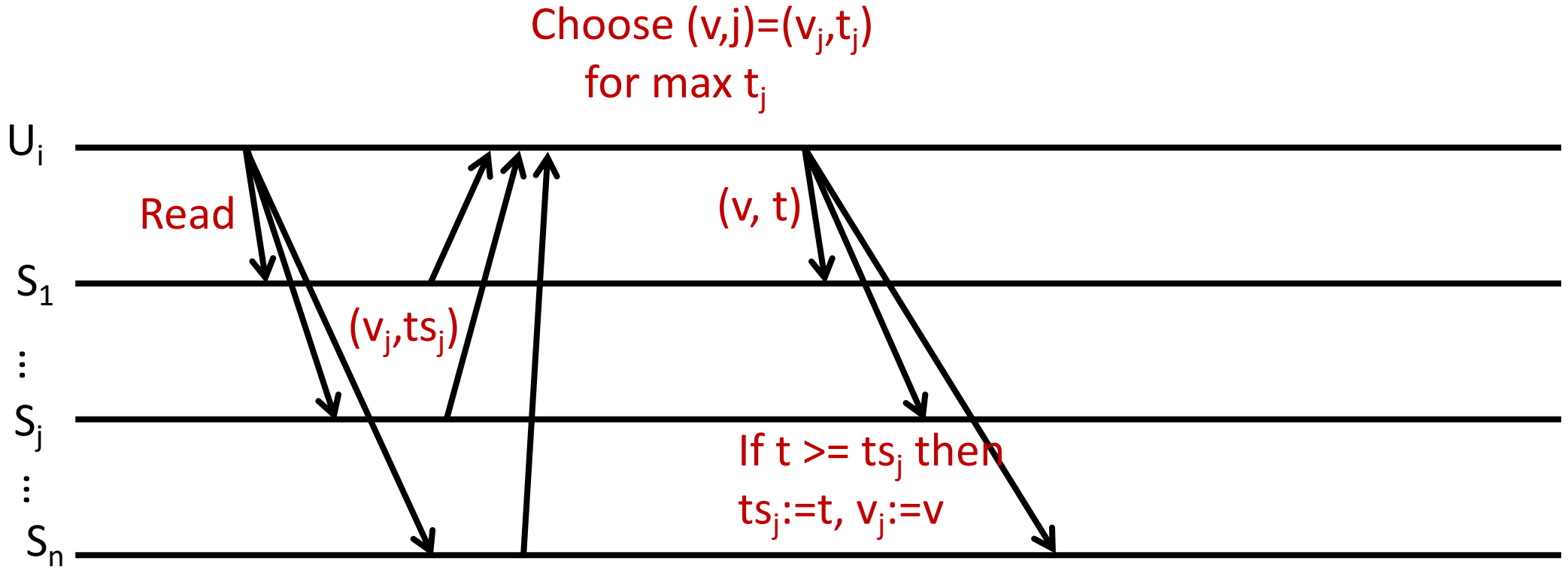
Read Operation

After received a majority of responses, choose the pair (v,t) such that t is the highest



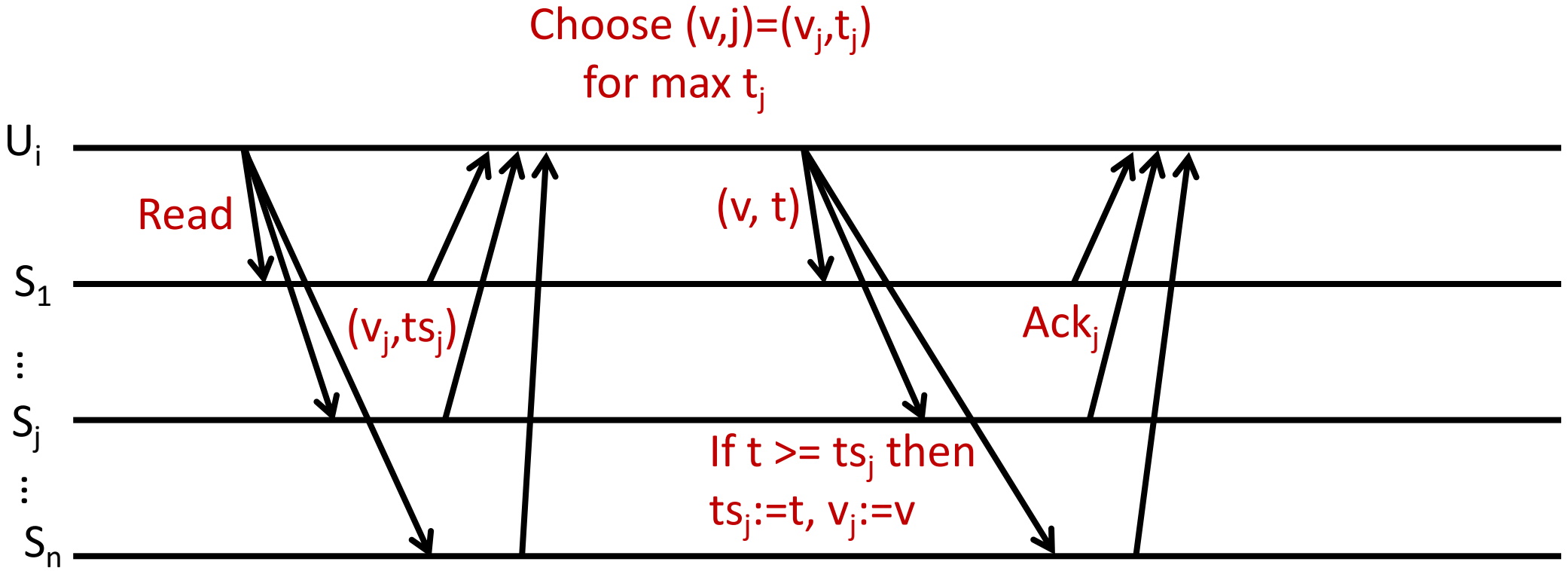
Read Operation

Send Write request with the chosen (v,t) pair to all servers

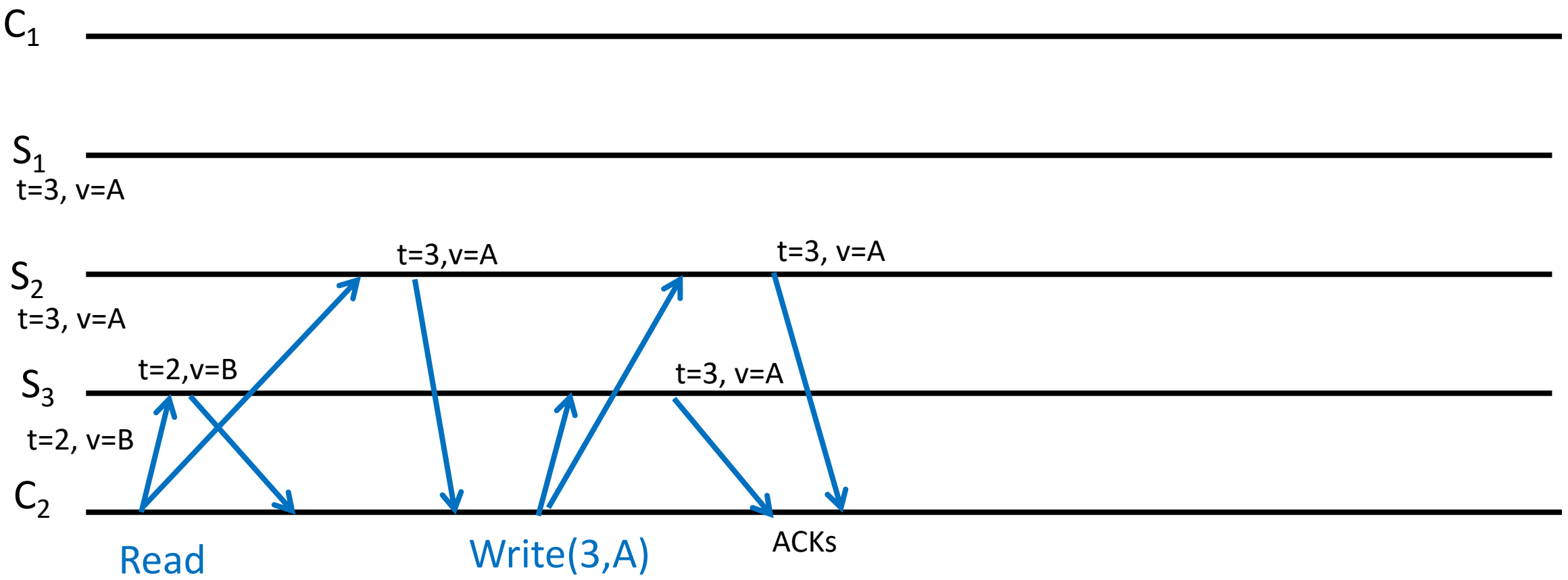


Read Operation

Read is completed when the client received a majority of ACKs



Read Updates Stale Servers

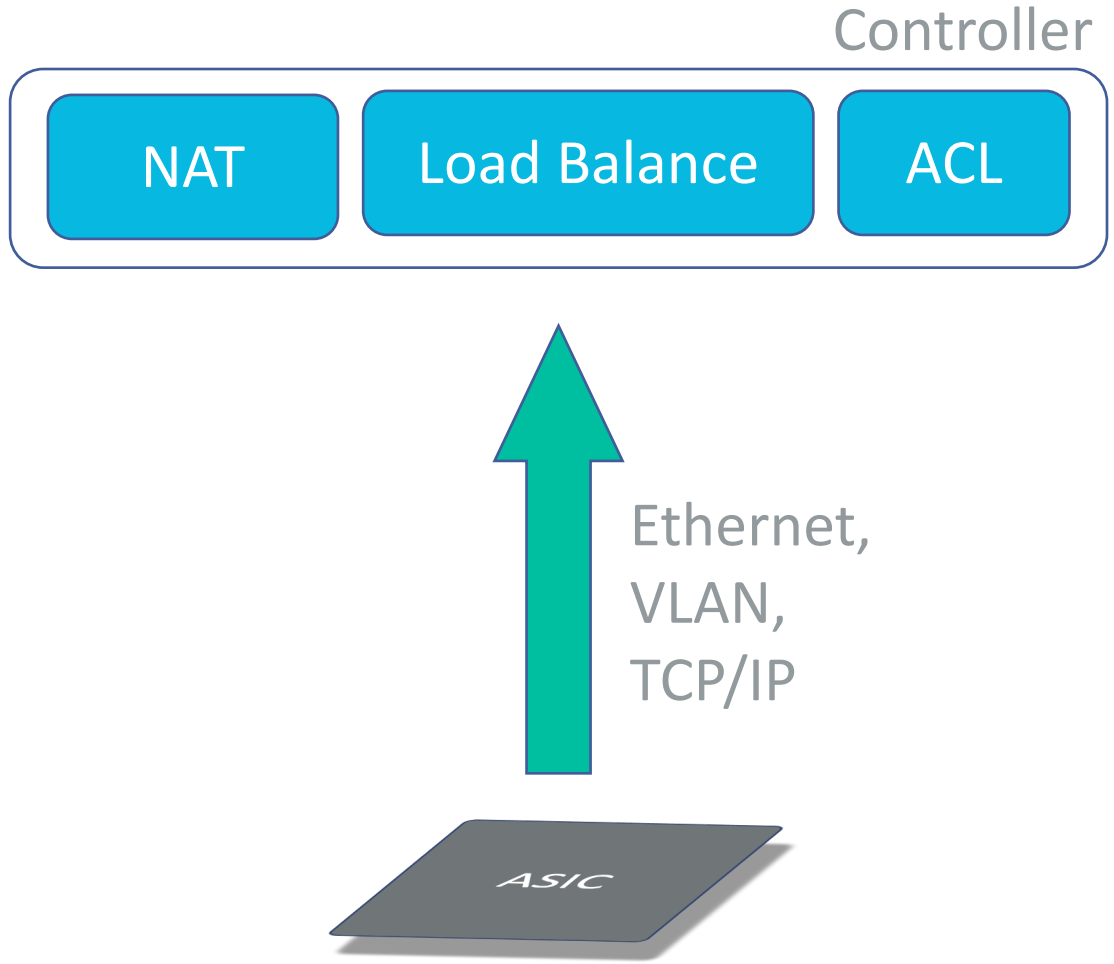


PISA™: Protocol Independent Switch Architecture

Traditional Switch Architecture

Fixed packet forwarding pipeline

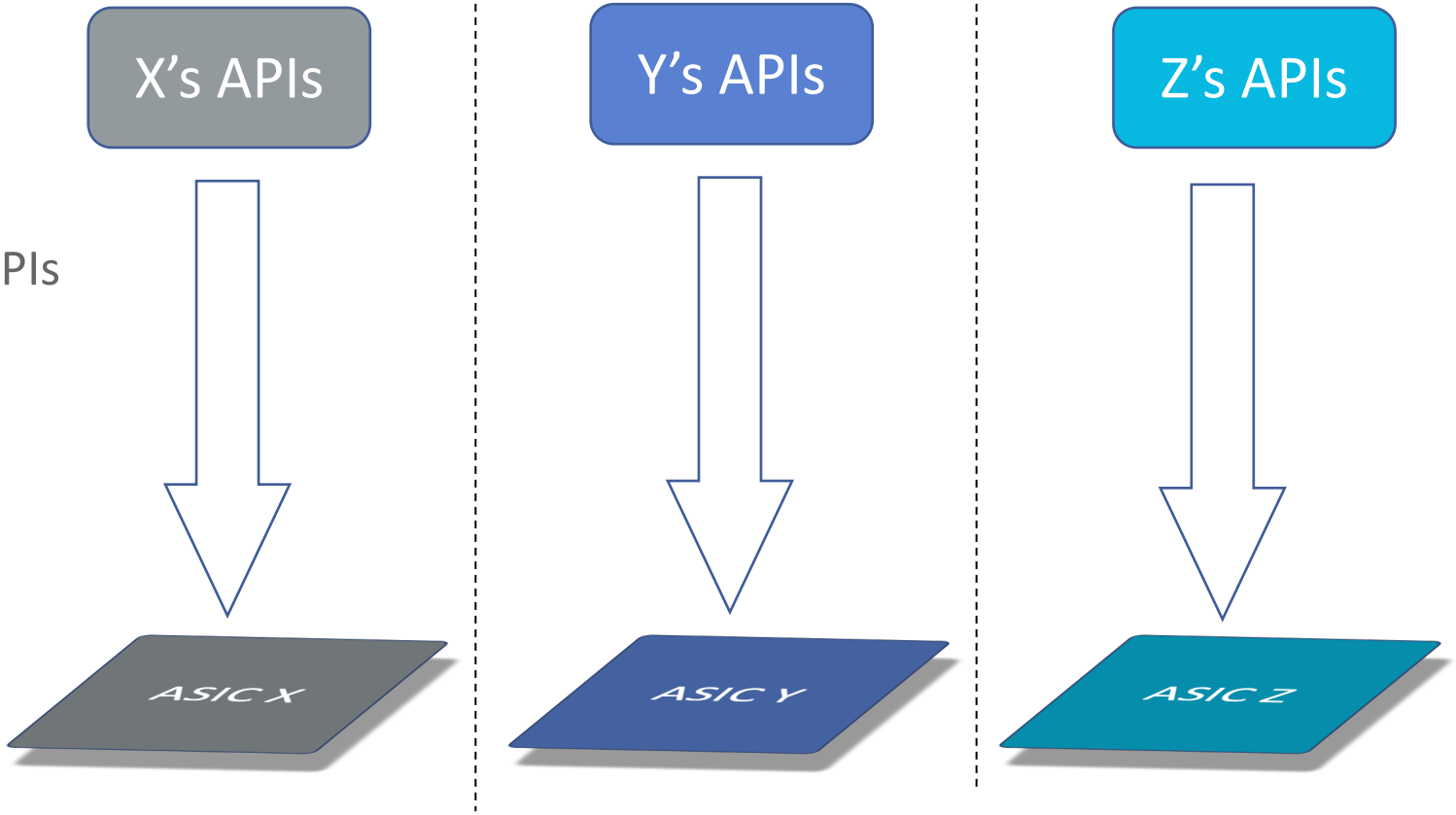
- Switch only knows how to process pre-defined packets in the ASIC
  - Dropped unknown packets
- It's expensive to fabricate ASIC chips
  - Limit innovation in networking
- Take decades to adopt new protocols
  - E.g., It took 20 years for IPv6 to become an Internet standard



Traditional Switch Architecture

Proprietary control and management

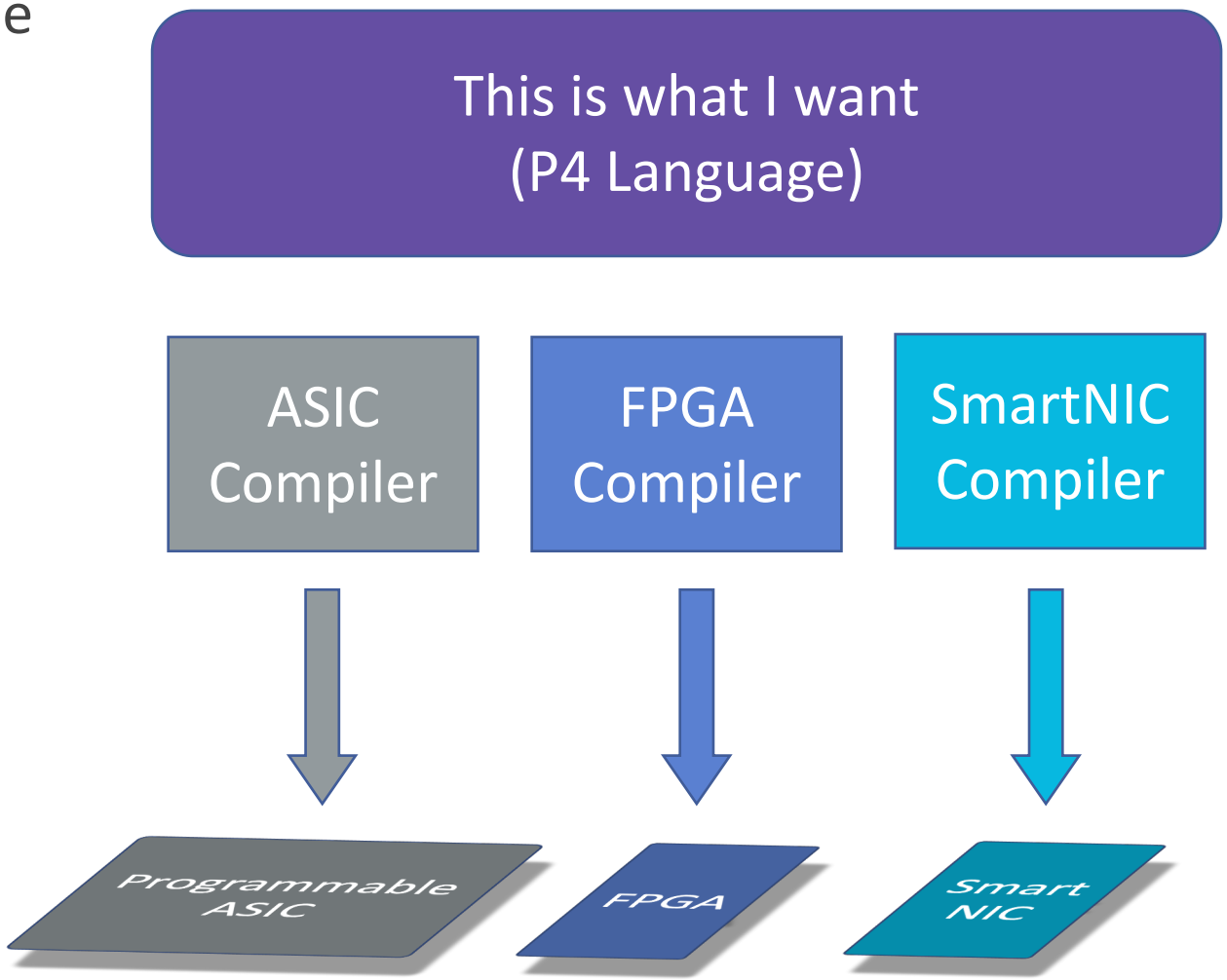
- Vendor lock-in
  - Proprietary tools/APIs are incompatible with those of competitors
- Difficult to integrate network devices from various vendors
  - Different vendors provide different APIs to manage their devices



PISA: Protocol Independent Switch Architecture

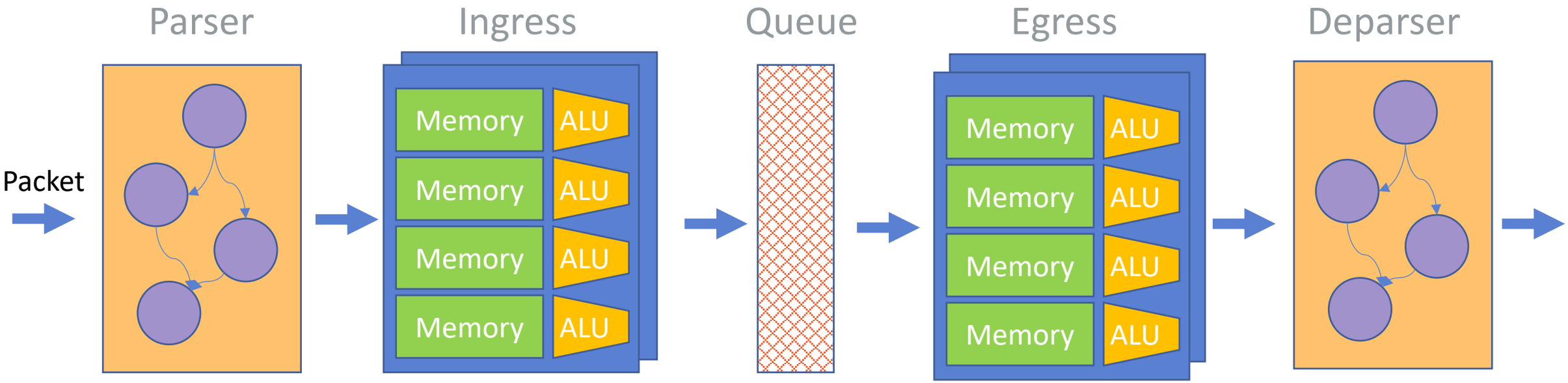
Abstract model of programmable switch architecture

- Programmable packet processing pipeline with ASIC-like performance
  - Develop and verify new protocols on a daily basis
- Standardized programming language
  - Write program once, run everywhere
- Vendor-agnostic Hardware
  - Barefoot Tofino™ ASIC, NetFPGA SUME, Netronome Agilio® SmartNICs, etc.



Programming Language for Network Data Plane

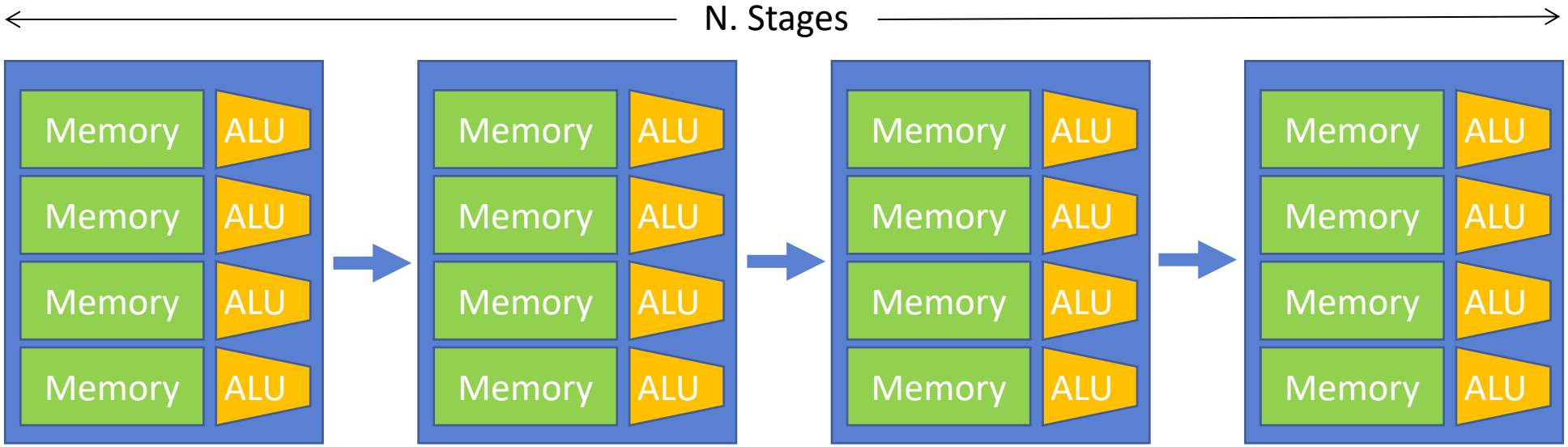
P4: Programming Protocol-Independent Packet Processors*



*https://www.sigcomm.org/sites/default/files/ccr/papers/2014/July/0000000-0000004.pdf

Constraints of Programmable ASICs

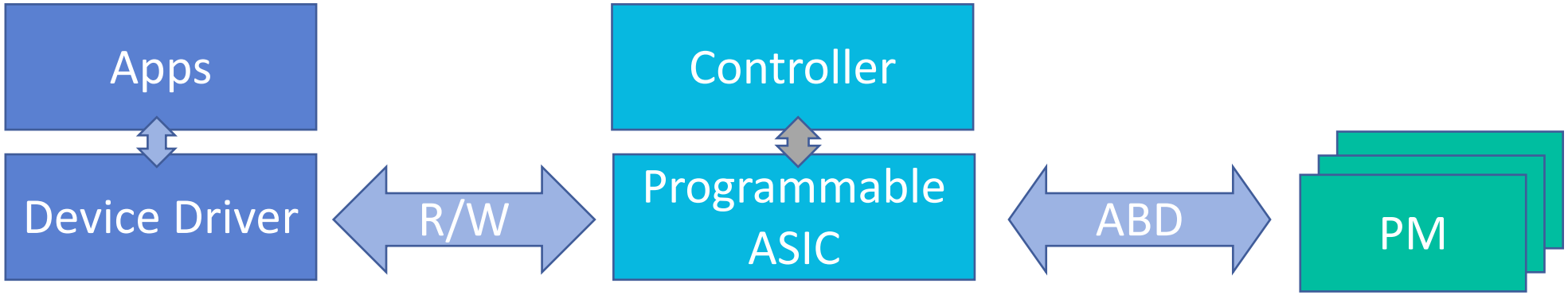
- **Memory**
  - the amount of memory available in each stage for stateful operations or match actions is limited
- **ALUs**
  - each stage of the pipeline has limited ALU units
- **Pipeline Depth**
  - there is a fixed number of stages per pipeline



System Design & Evaluation

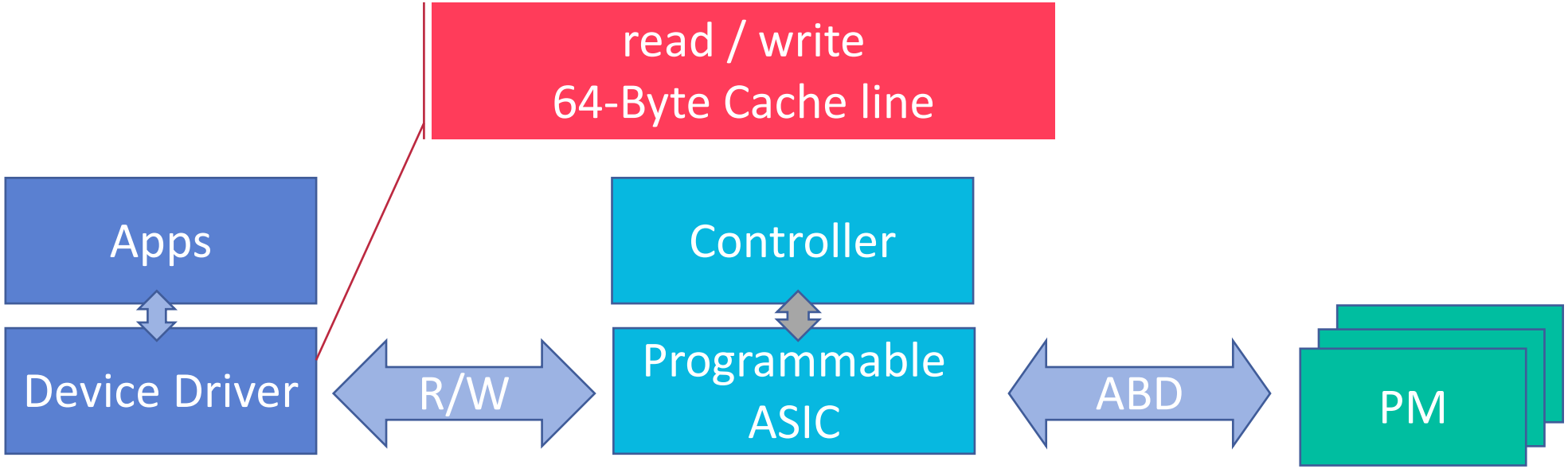
System Design

- Client: a custom Linux kernel device driver
- A set of persistent memory instances
- A programmable switch runs ABD protocol



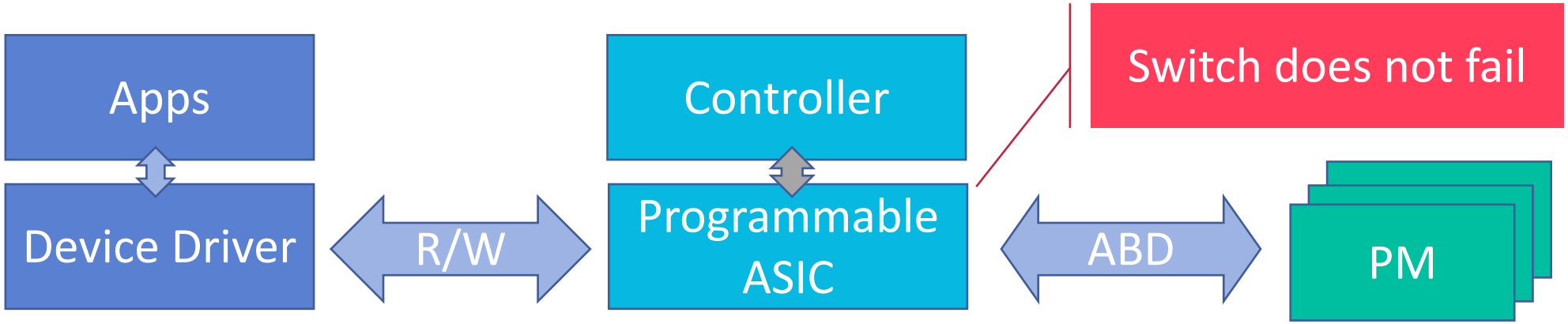
Memory Controller

- A character device driver implements **mmap** APIs
  - Applications open the device for read/write
  - A page fault will trigger a read/write to remote memory
  - The device driver waits for a response from remote memories before return data or acknowledgement to the applications



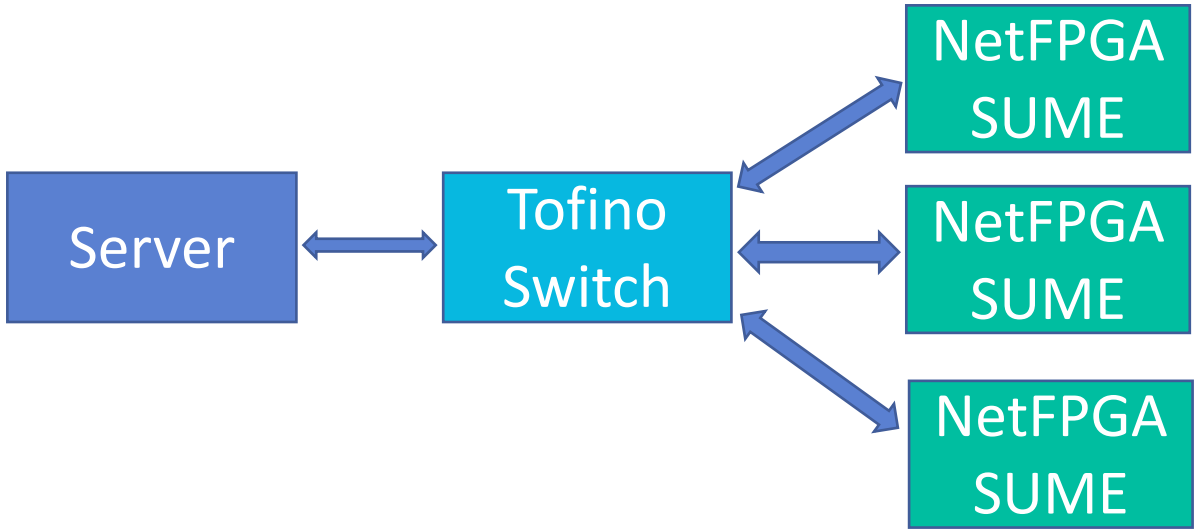
Network Fabric

- Assumption: switches do not fail
  - The mean time to failure for memory is significantly shorter than the mean time to failure for switches
- Implementation:
  - Generalize the ABD protocol to support multiple cache lines
  - Store a timestamp per cache line
  - Forward packets based on Ethernet MAC



Evaluation

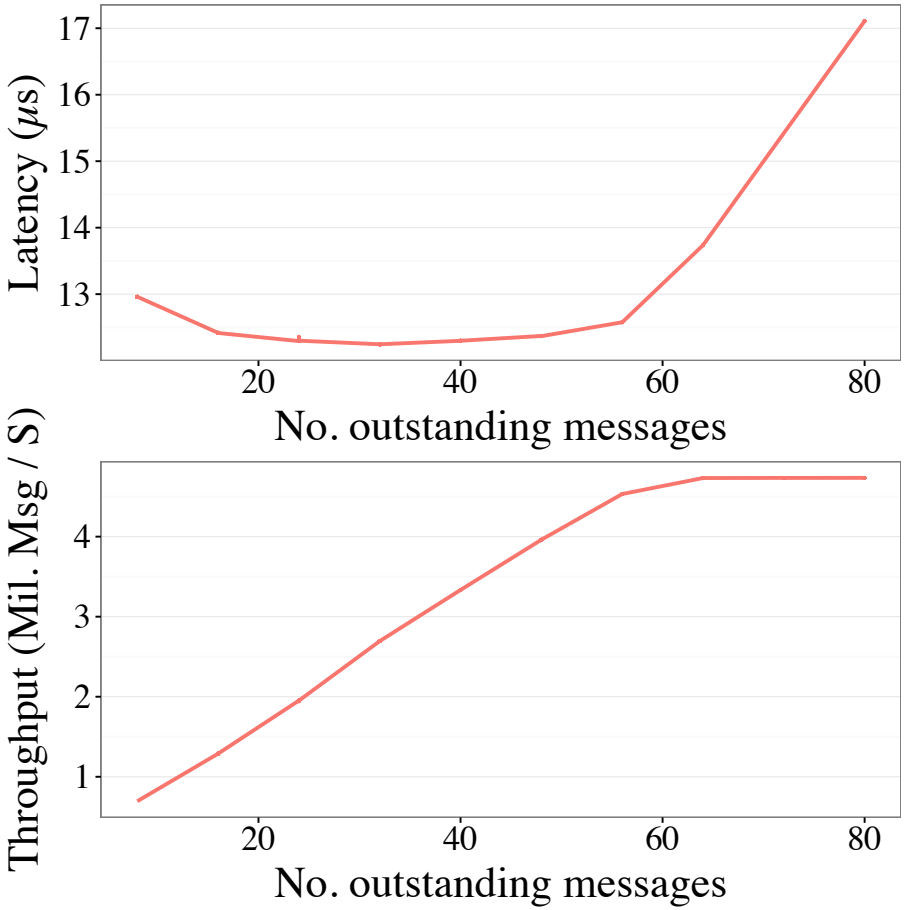
- Barefoot Tofino Switch 32X
  - Configured 10GbE per port
  - P4-14 code compiled with Barefoot Capilano
- NetFPGA SUME
  - P4-16 code compiled with P4-NetFPGA compiler
- SuperMicro® Server (traffic generator)
  - dual-socket Intel® Xeon® E5-2603 CPUs
  - 16GB of 1600MHz DDR4
  - Intel 82599 10Gbps NIC



Preliminary Result

Latency of write cache lines to the replicated remote memories

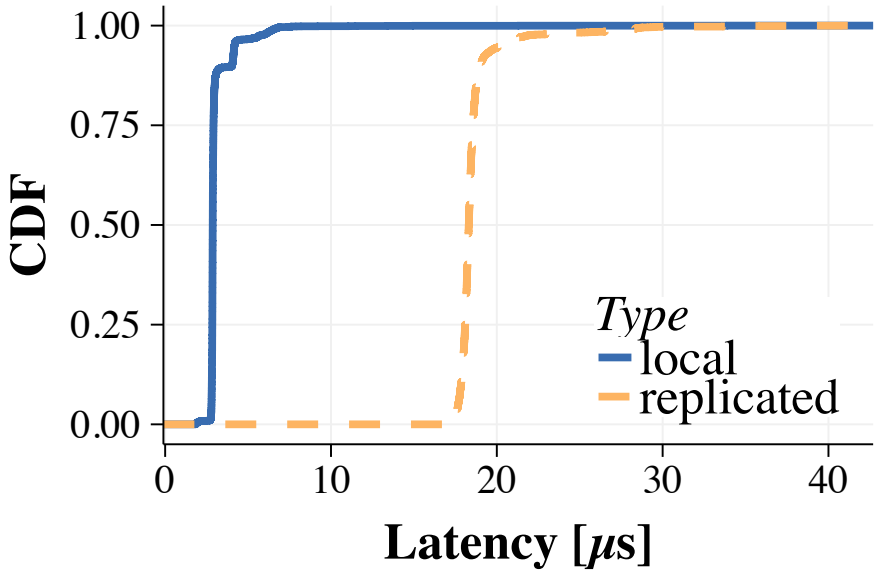
- Issue increasing number of writes
  - Write to different pages
  - Measure latency for 100K requests
- **Software** device driver is saturated
  - Cap throughput is around 4.4 MMsg/s
  - Latency increase beyond saturation point



Preliminary Result

Latency CDF read/write cache lines from local memory and from the replicated remote memories

- Server emulates memory controller
  - implement a character device driver
  - ‘mmap’ file into memory
  - Handle page faults by sending and receiving network packets
  - Measure latency for 100K requests
- Read/write cdf latency
  - Local memory 3µs vs. replicated remote 18µs
- Traditional replicated storage system
  - Latency >100µs

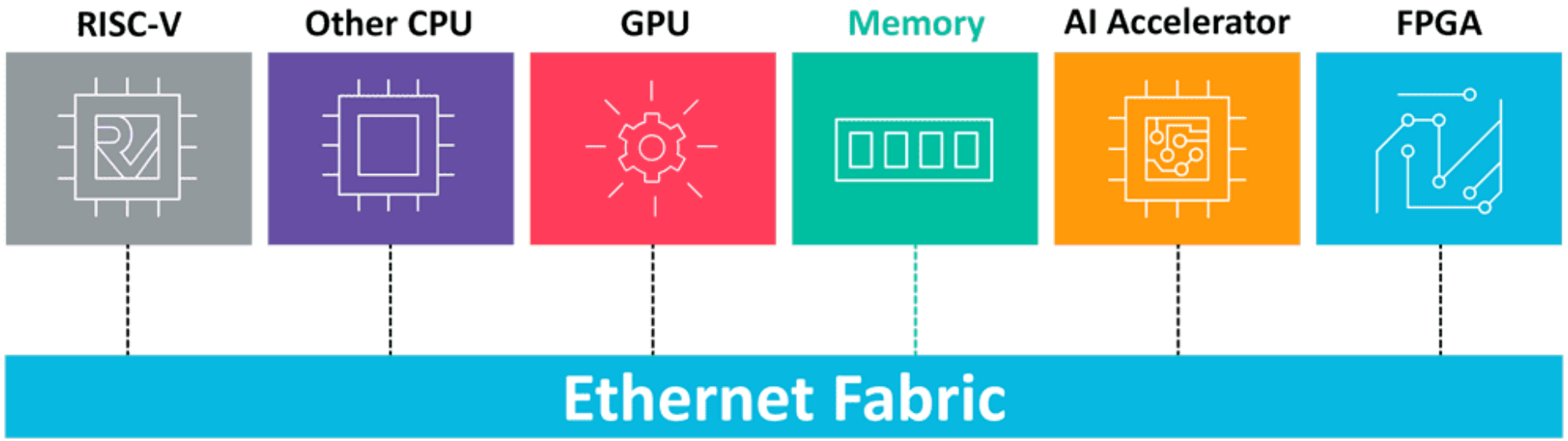


Conclusions

- Persistent memory can transform the memory hierarchy
- In-Network consensus helps solve a critical challenge of persistent memory
- Initial experiments demonstrate order-of-magnitude faster than traditional replicated system and shows great promise as scalable memory

Open Questions

- How do we preserve same liveness guarantee if switch fails or packet lost?
  - Fail-over to a backup switch
- What is the interplay between cache coherency and consistency?
  - OmniXtend™ An open standard Cache Coherent Fabric Interface repository



Huynh Tu Dang

Principal Research Engineer

E-mail: tu.dang@wdc.com
homepage: http://tudang.github.io

Western Digital®

Architecting Data Infrastructure for the Zettabyte Age