The Paxos Register

Harry C. Li, Allen Clement, Amitanand Aiyer, Lorenzo Alvisi

The University of Texas at Austin

Consensus & Paxos

- Validity
- Integrity
- Agreement
- Termination

Consensus & Paxos

Part-time Parliament [Lamport '98]

- Validity
- Integrity
- Agreement
- Termination

Consensus & Paxos

Part-time Parliament [Lamport '98]

Safety

- Validity
- Integrity
- Agreement

- Termination

Liveness

Consensus & Paxos

Part-time Parliament [Lamport '98]
Frangipani [Thekkath et al. '97]

Safety

- Validity
- Integrity
- Agreement

- Termination

Liveness

Consensus & Paxos

Part-time Parliament [Lamport '98]
Frangipani [Thekkath et al. '97]
Byzantine Paxos [Castro & Liskov '99]

Safety

- Validity
- Integrity
- Agreement

- Termination

Liveness

Consensus & Paxos

Safety
- Validity
- Integrity
- Agreement

Termination

Liveness

Part-time Parliament	[Lamport '98]
Frangipani	[Thekkath et al. '97]
Byzantine Paxos	[Castro & Liskov '99]
Disk Paxos	[Gafni & Lamport '00]

Consensus & Paxos

Safety
- Validity
- Integrity
- Agreement

Termination

Liveness

Part-time Parliament	[Lamport '98]
Frangipani	[Thekkath et al. '97]
Byzantine Paxos	[Castro & Liskov '99]
Disk Paxos	[Gafni & Lamport '00]
Deconstructing Paxos	[Boichat et al. '01]
Reconstructing Paxos	[Boichat et al. '01]

Consensus & Paxos

Safety
- Validity
- Integrity
- Agreement

Termination

Liveness

Part-time Parliament	[Lamport '98]
Frangipani	[Thekkath et al. '97]
Byzantine Paxos	[Castro & Liskov '99]
Disk Paxos	[Gafni & Lamport '00]
Deconstructing Paxos	[Boichat et al. '01]
Reconstructing Paxos	[Boichat et al. '01]
Active Disk Paxos	[Chockler & Malkhi '02]

Consensus & Paxos

Safety
- Validity
- Integrity
- Agreement

Termination

Liveness

Part-time Parliament	[Lamport '98]
Frangipani	[Thekkath et al. '97]
Byzantine Paxos	[Castro & Liskov '99]
Disk Paxos	[Gafni & Lamport '00]
Deconstructing Paxos	[Boichat et al. '01]
Reconstructing Paxos	[Boichat et al. '01]
Active Disk Paxos	[Chockler & Malkhi '02]
Agreement & Execution	[Yin et al. '03]

Consensus & Paxos

Safety
- Validity
- Integrity
- Agreement

Termination

Liveness

Part-time Parliament	[Lamport '98]
Frangipani	[Thekkath et al. '97]
Byzantine Paxos	[Castro & Liskov '99]
Disk Paxos	[Gafni & Lamport '00]
Deconstructing Paxos	[Boichat et al. '01]
Reconstructing Paxos	[Boichat et al. '01]
Active Disk Paxos	[Chockler & Malkhi '02]
Agreement & Execution	[Yin et al. '03]
Byzantine Disk Paxos	[Abraham et al. '04]

Consensus & Paxos

Safety
- Validity
- Integrity
- Agreement

Termination

Liveness

Part-time Parliament	[Lamport '98]
Frangipani	[Thekkath et al. '97]
Byzantine Paxos	[Castro & Liskov '99]
Disk Paxos	[Gafni & Lamport '00]
Deconstructing Paxos	[Boichat et al. '01]
Reconstructing Paxos	[Boichat et al. '01]
Active Disk Paxos	[Chockler & Malkhi '02]
Agreement & Execution	[Yin et al. '03]
Byzantine Disk Paxos	[Abraham et al. '04]
Fast Byzantine Paxos	[Martin & Alvisi '05]
Fast Paxos	[Lamport '05]

Consensus & Paxos

Safety

- Validity
- Integrity
- Agreement
- Termination

Liveness

Part-time Parliament	[Lamport '98]
Frangipani	[Thekkath et al. '97]
Byzantine Paxos	[Castro & Liskov '99]
Disk Paxos	[Gafni & Lamport '00]
Deconstructing Paxos	[Boichat et al. '01]
Reconstructing Paxos	[Boichat et al. '01]
Active Disk Paxos	[Chockler & Malkhi '02]
Agreement & Execution	[Yin et al. '03]
Byzantine Disk Paxos	[Abraham et al. '04]
Fast Byzantine Paxos	[Martin & Alvisi '05]
Fast Paxos	[Lamport '05]
Hybrid Quorums	[Cowling et al. '06]
Chubby	[Burrows '06]
Zyzzyva	[Kotla et al. '07]

What Makes Paxos Paxos?

What Makes Paxos Paxos?

- Guarantee safety
- Provide liveness during periods of synchrony
- Use leaders to coordinate quorums

What Makes Paxos Paxos?

- Guarantee safety
- Provide liveness during periods of synchrony
- Use leaders to coordinate quorums

Is that it?

Understanding Paxos

Abstract Paxos	[Lampson '01]
◇Register & ◇Leader	[Boichat et al. '01]
Ranked Register	[Chockler & Malkhi '05]
Alpha of Indulgent Consensus	[Guerraoui & Raynal '07]

Understanding Paxos

Crash and Byzantine

Abstract Paxos	[Lampson '01]
◇Register & ◇Leader	[Boichat et al. '01]
Ranked Register	[Chockler & Malkhi '05]
Alpha of Indulgent Consensus	[Guerraoui & Raynal '07]

Understanding Paxos

Crash and Byzantine

Abstract Paxos	[Lampson '01]
♢Register & ♢Leader	[Boichat et al. '01]
Ranked Register	[Chockler & Malkhi '05]
Alpha of Indulgent Consensus	[Guerraoui & Raynal '07]

Register-based

Understanding Paxos

Crash and Byzantine

Abstract Paxos	[Lampson '01]
♢Register & ♢Leader	[Boichat et al. '01]
Ranked Register	[Chockler & Malkhi '05]
Alpha of Indulgent Consensus	[Guerraoui & Raynal '07]

Register-based

No register-based description encompassing both crash and Byzantine variants

Paxos Register

Paxos Register

- Captures crash and Byzantine Paxos variants

Paxos Register

- Captures crash and Byzantine Paxos variants
- Focuses on most difficult part of Paxos

Paxos Register

- Captures crash and Byzantine Paxos variants
- Focuses on most difficult part of Paxos
- Implements a write-once register

Paxos Register

- Captures crash and Byzantine Paxos variants
- Focuses on most difficult part of Paxos
- Implements a write-once register

read ⟶ write

Paxos Register

- Captures crash and Byzantine Paxos variants
- Focuses on most difficult part of Paxos
- Implements a write-once register

read ⟶ write

token

Paxos Register

- Captures crash and Byzantine Paxos variants
- Focuses on most difficult part of Paxos
- Implements a write-once register

read ⟶ write

token

- Describes Classic, Byzantine, and FaB Paxos
- Leads to Byzantine Paxos using secret sharing

Outline

- Motivation
- Paxos Register specification
- Crash implementation
- Byzantine implementation

Consensus & Paxos Register

world time ⟶

Consensus & Paxos Register

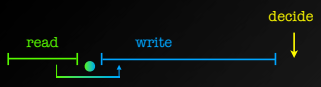
read

world time ⟶

Consensus & Paxos Register

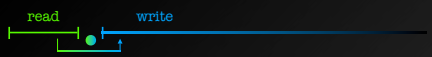
world time

Consensus & Paxos Register



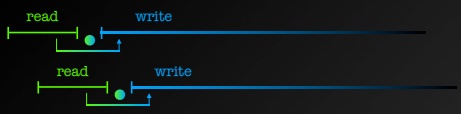
world time

Consensus & Paxos Register



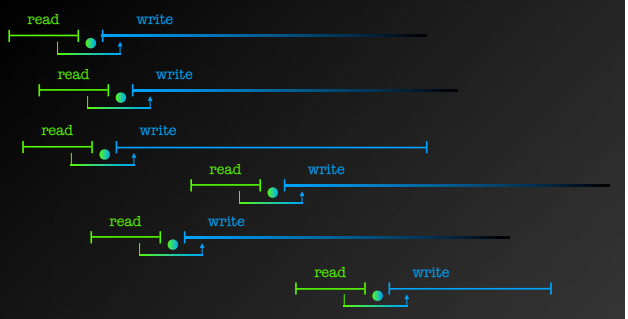
world time

Consensus & Paxos Register



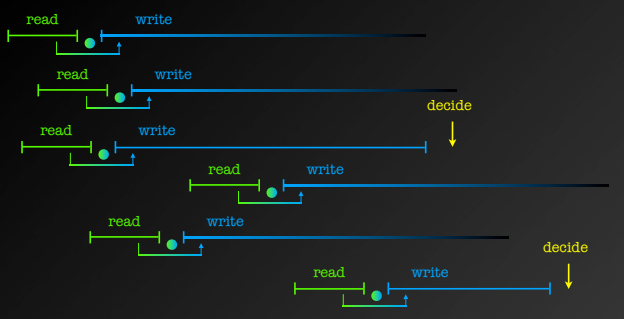
world time

Consensus & Paxos Register



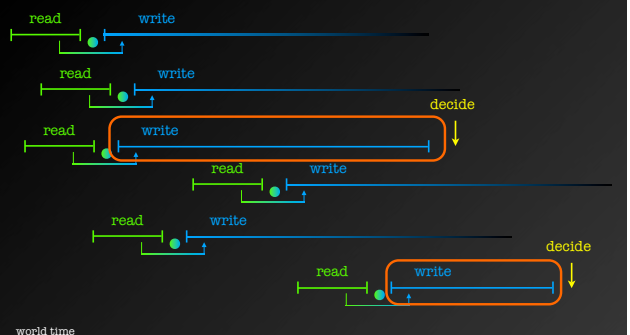
world time

Consensus & Paxos Register



world time

Consensus & Paxos Register



Details

world time

Details

read
token (v=⊥, ts=0)

world time

Details

read
token (v=⊥, ts=0)

write
v=a, token = (⊥,0)

write starts as partial

world time

Details

read
token (v=⊥, ts=0)

write
v=a, token = (⊥,0)

write starts as partial
partial write may become visible

world time

Details

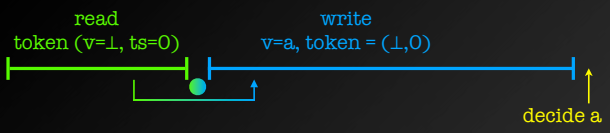
read
token (v=⊥, ts=0)

write
v=a, token = (⊥,0)

write starts as partial
partial write may become visible
visible write may become total

world time

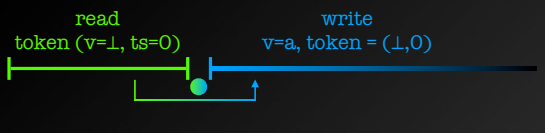
Details



write starts as partial
partial write may become visible
visible write may become total

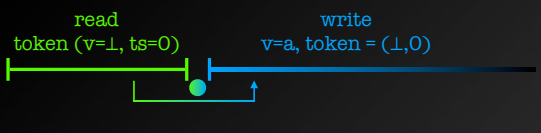
world time

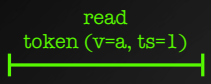
Details



world time

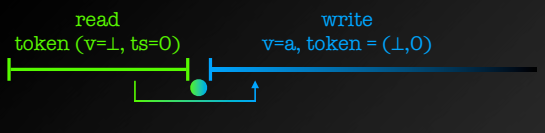
Details

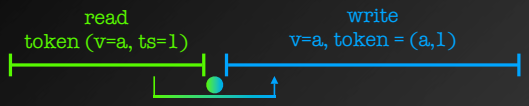




world time

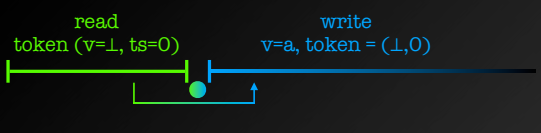
Details

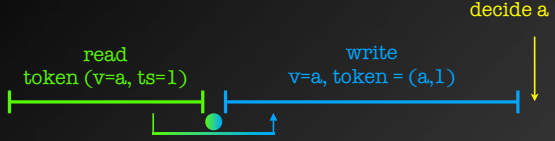




world time

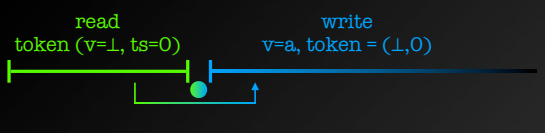
Details

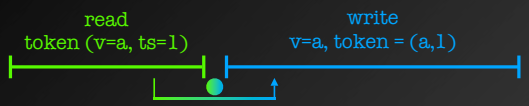




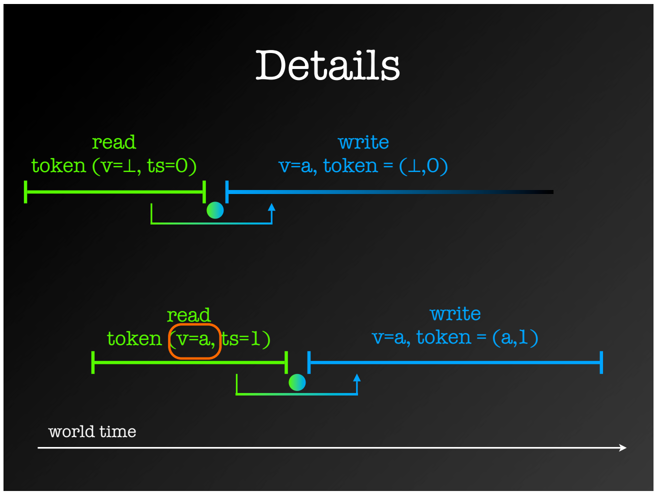
world time

Details

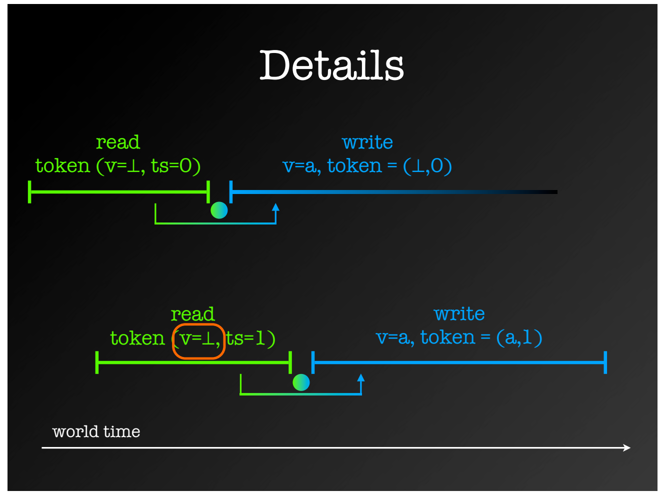




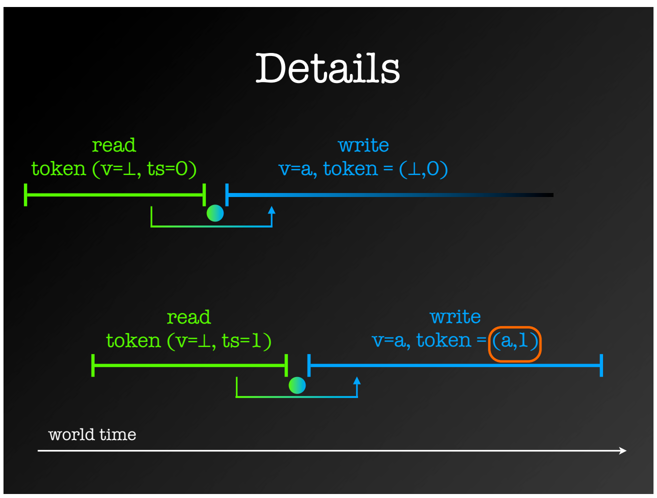
world time



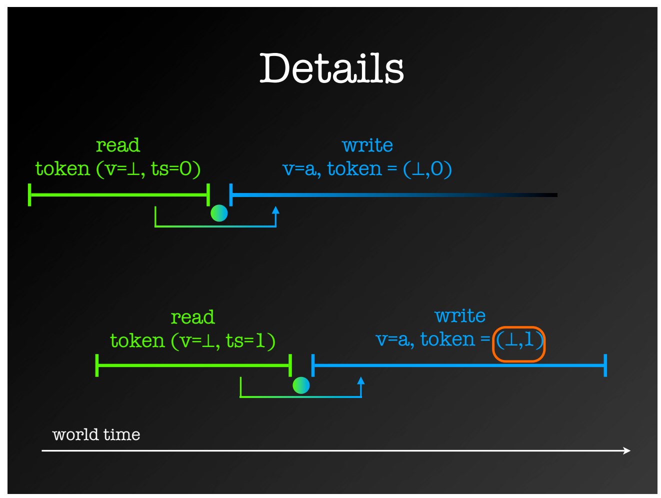
11



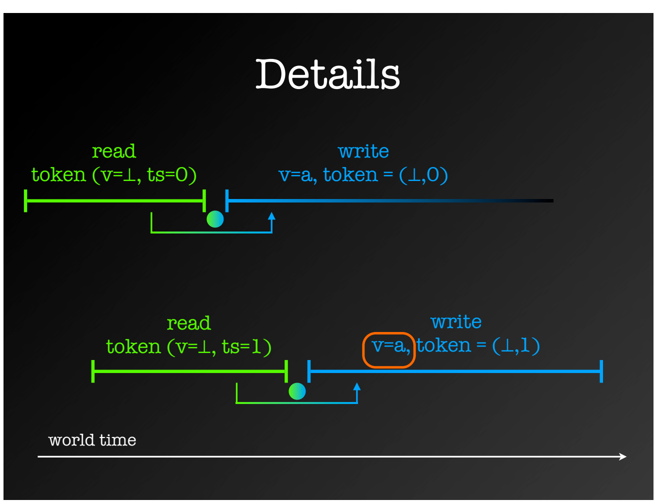
11



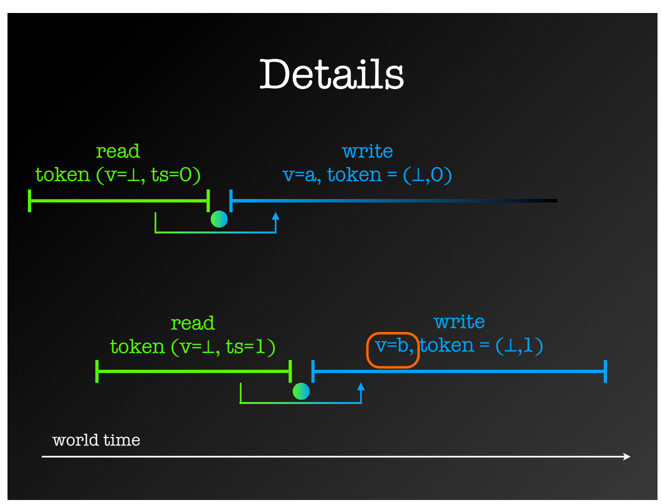
11



11

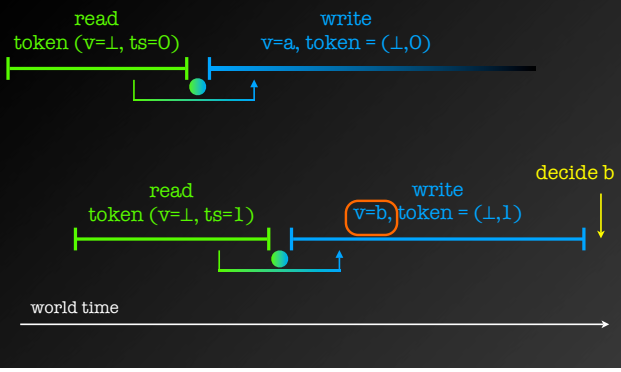


11

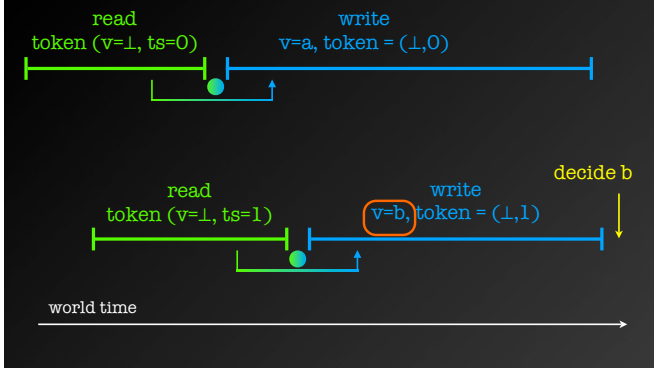


11

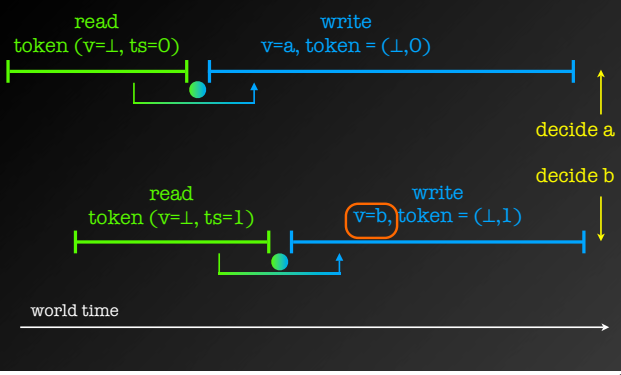
Details



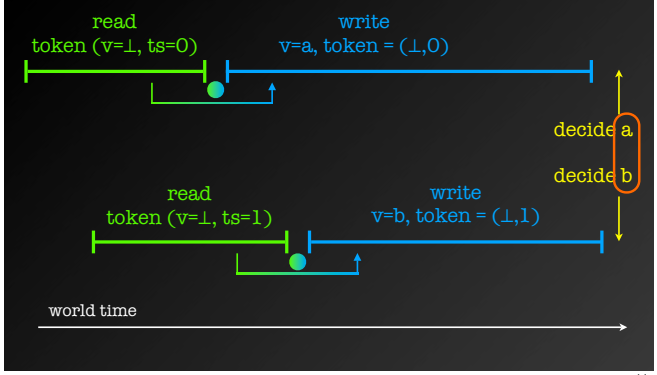
Details



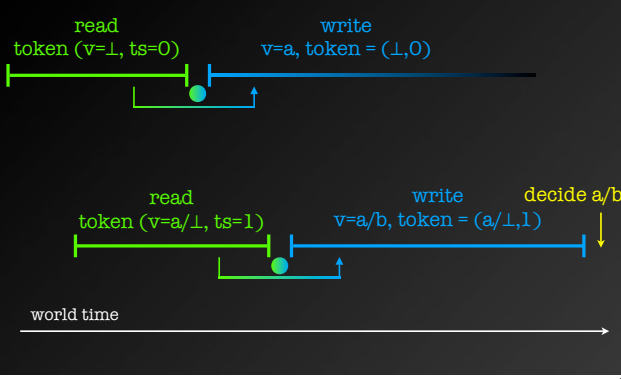
Details



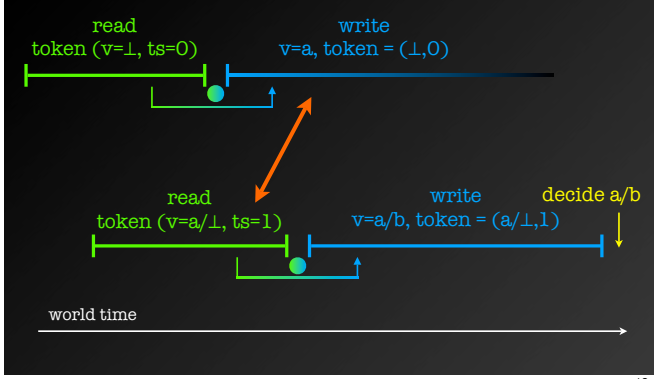
Details



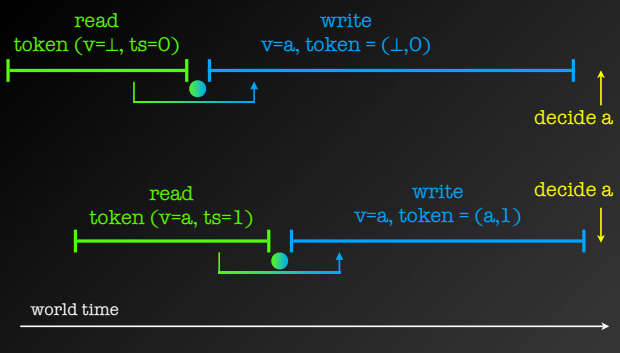
Concurrent Operations



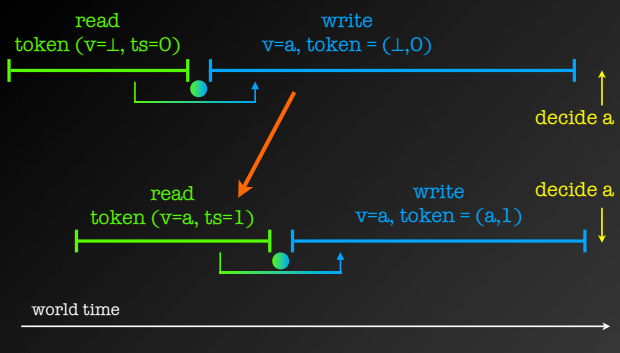
Concurrent Operations



Almost Regular



Almost Regular



Consistency Semantics

read returns last written value or any value being concurrently written

Consistency Semantics

read returns last written value or any value being concurrently written

- Paxos Register redefines 'concurrent'

Consistency Semantics

read returns last written value or any value being concurrently written

- Paxos Register redefines 'concurrent'
- Regular w.r.t. modified partial order
  1. Writes ordered by increasing timestamp
  2. Total write precedes higher timestamped read
  3. Read precedes higher or same timestamped write

Consistency Semantics

read returns last written value or any value being concurrently written

- Paxos Register redefines 'concurrent'
- Regular w.r.t. modified partial order
  1. Writes ordered by increasing timestamp
  2. Total write precedes higher timestamped read
  3. Read precedes higher or same timestamped write

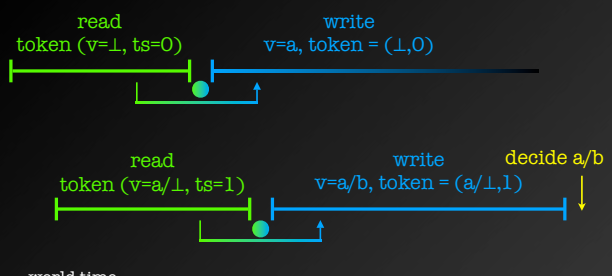
Modified Partial Order

2. Total write precedes higher timestamped read

world time

Modified Partial Order

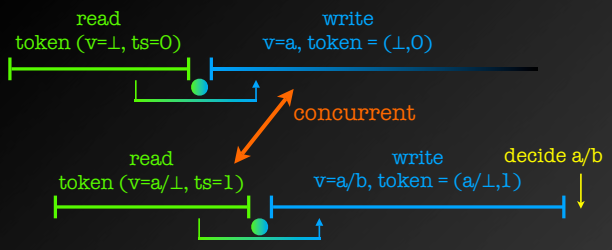
2. Total write precedes higher timestamped read



world time

Modified Partial Order

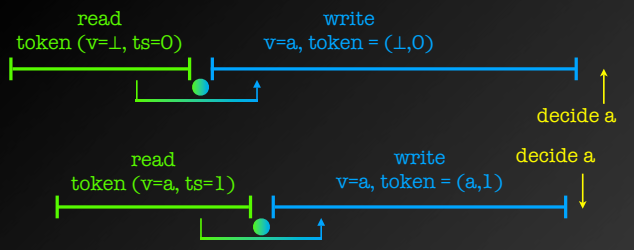
2. Total write precedes higher timestamped read



world time

Modified Partial Order

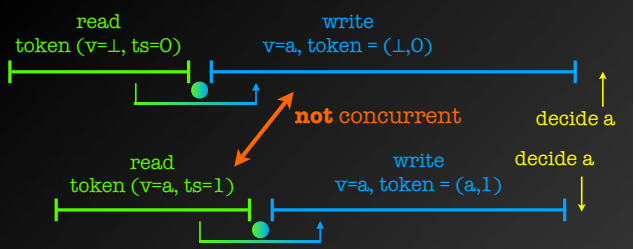
2. Total write precedes higher timestamped read



world time

Modified Partial Order

2. Total write precedes higher timestamped read



world time

Write-Once Register

Write-Once Register

Theorem: All total writes write the same value.

Write-Once Register

Theorem: All total writes write the same value.

- Paxos variants share high-level protocol

Write-Once Register

Theorem: All total writes write the same value.

- Paxos variants share high-level protocol
- Paxos variants use common abstractions
  - Timestamps, tokens
  - Partial, total, and visible writes

Write-Once Register

Theorem: All total writes write the same value.

- Paxos variants share high-level protocol
- Paxos variants use common abstractions
  - Timestamps, tokens
  - Partial, total, and visible writes
  - Implementations differ

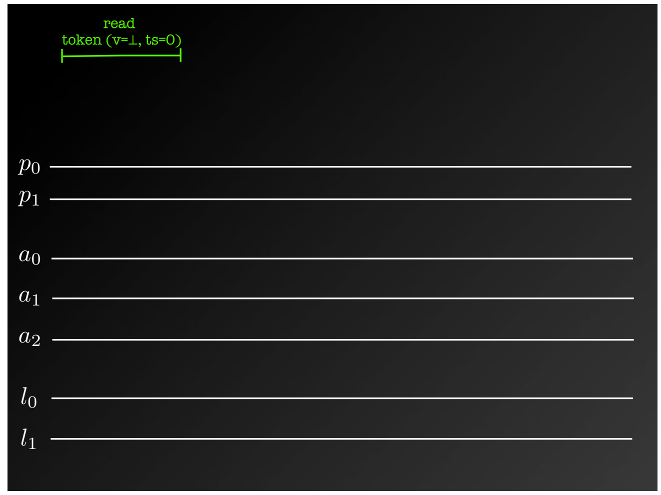
Outline

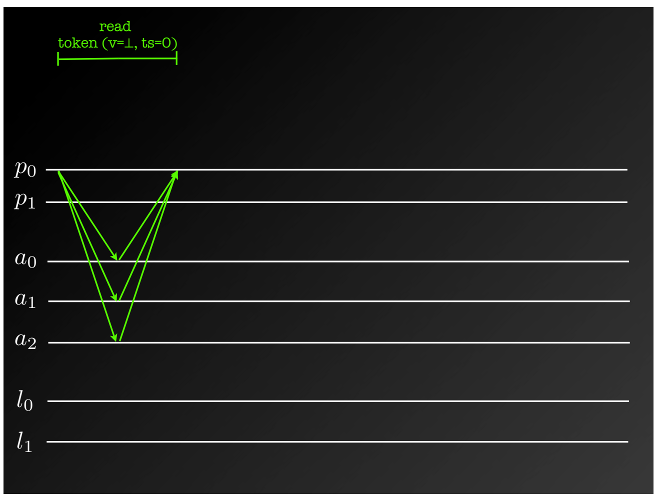
- Motivation
- Paxos Register specification
- Crash Implementation
- Byzantine Implementation

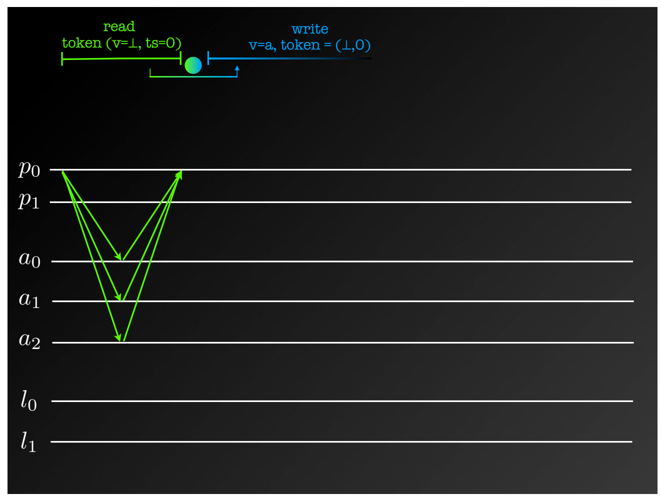
Crash Model

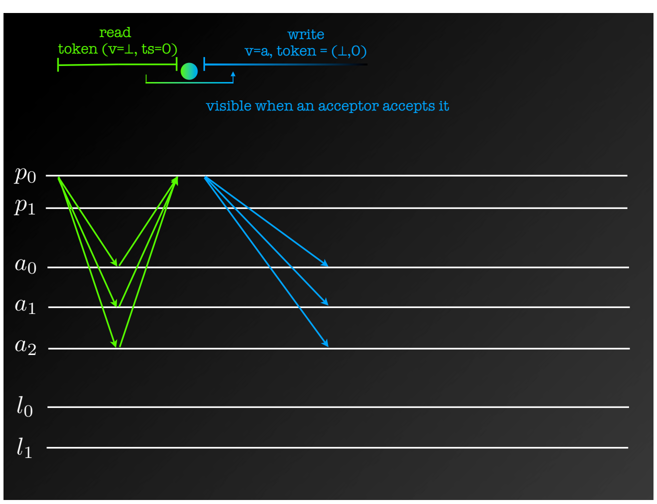
- Asynchronous system
- Proposers, acceptors, learners
- At least one proposer correct
- Majority of acceptors correct

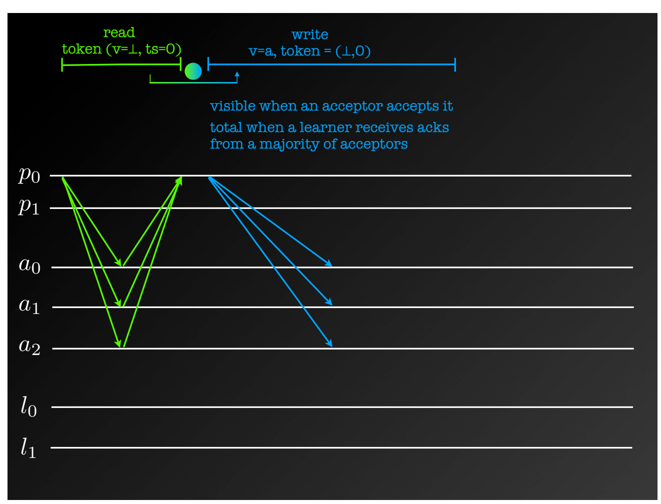

p0
p1
a0
a1
a2
l0
l1

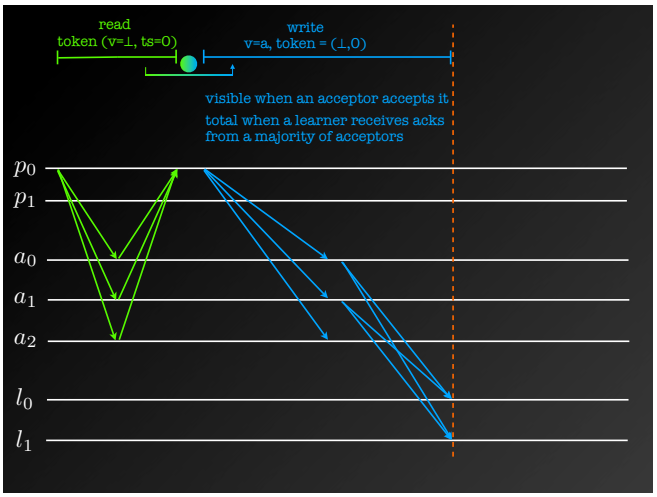

read
token (v=⊥, ts=0)
p0
p1
a0
a1
a2
l0
l1


read
token (v=⊥, ts=0)
p0
p1
a0
a1
a2
l0
l1

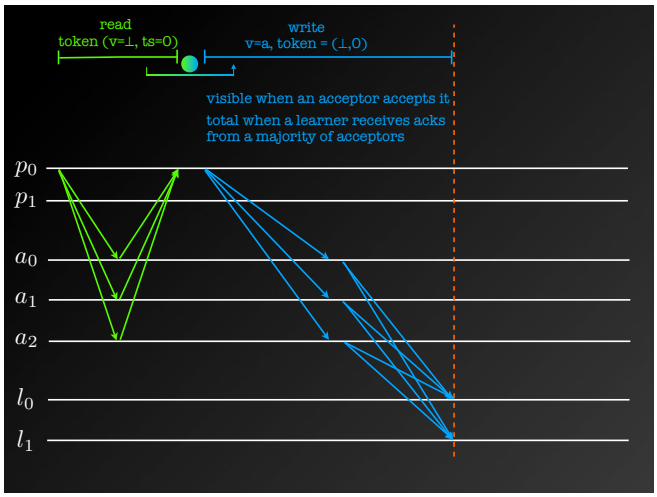

read
token (v=⊥, ts=0)
write
v=a, token = (⊥,0)
p0
p1
a0
a1
a2
l0
l1


read
token (v=⊥, ts=0)
write
v=a, token = (⊥,0)
visible when an acceptor accepts it
p0
p1
a0
a1
a2
l0
l1

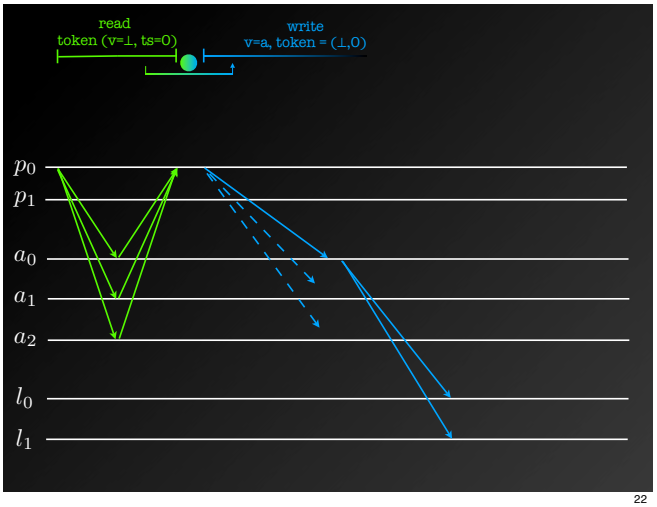

read
token (v=⊥, ts=0)
write
v=a, token = (⊥,0)
visible when an acceptor accepts it
total when a learner receives acks
from a majority of acceptors
p0
p1
a0
a1
a2
l0
l1



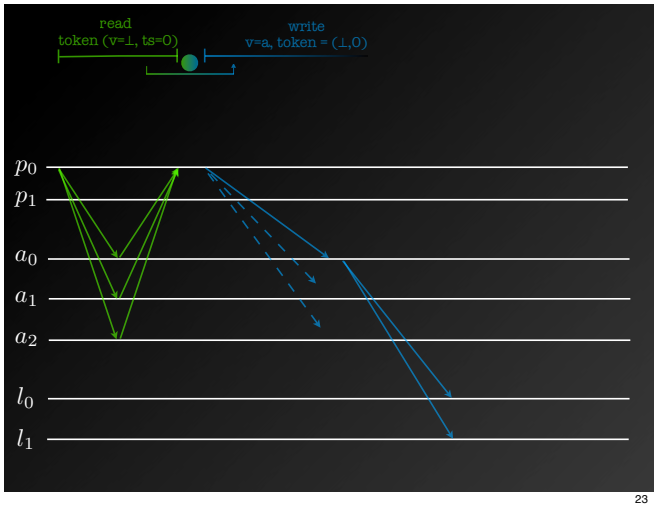
21



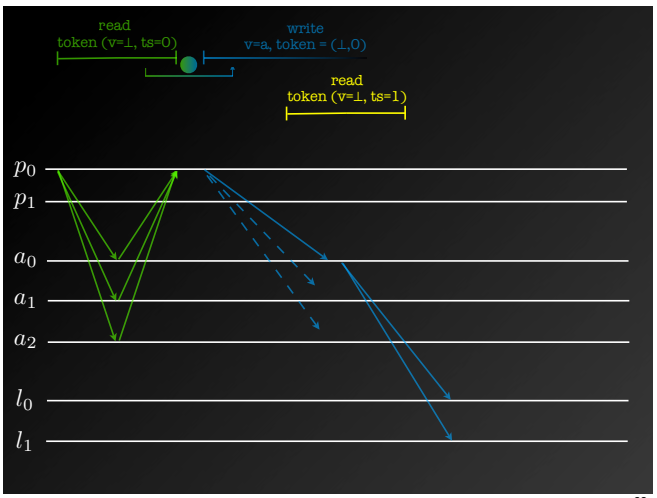
21



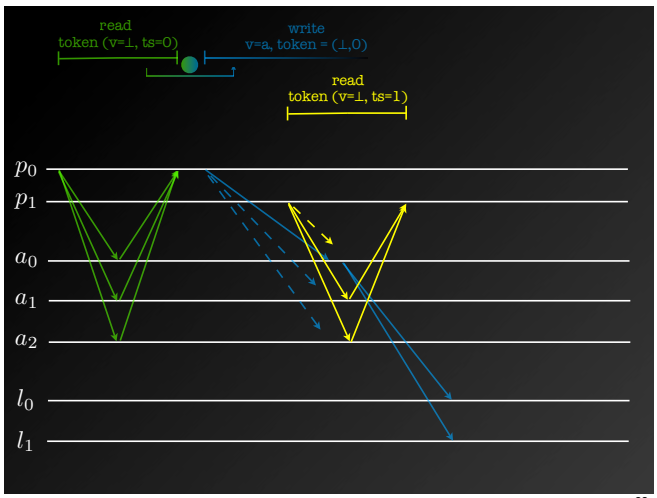
22



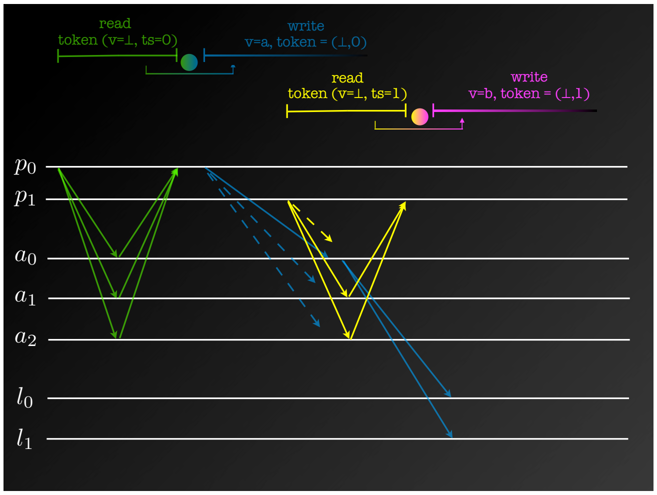
23

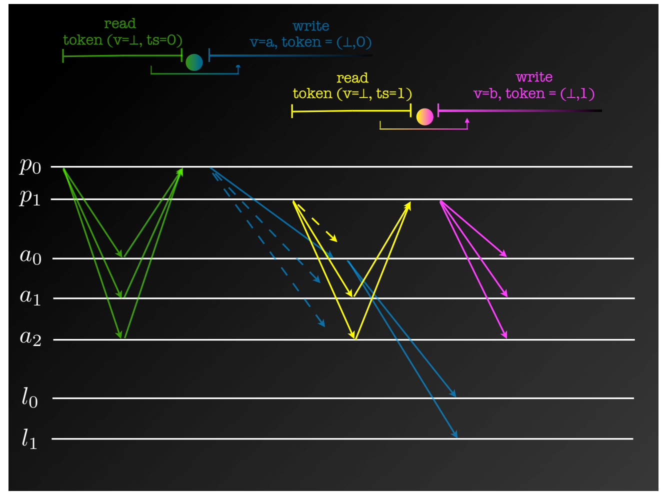


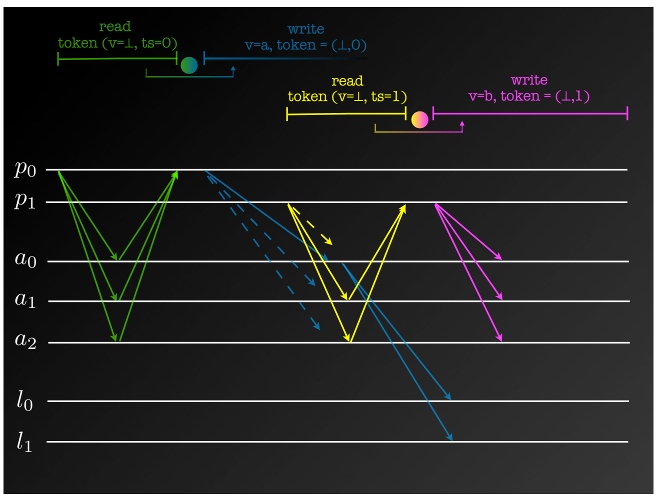
23

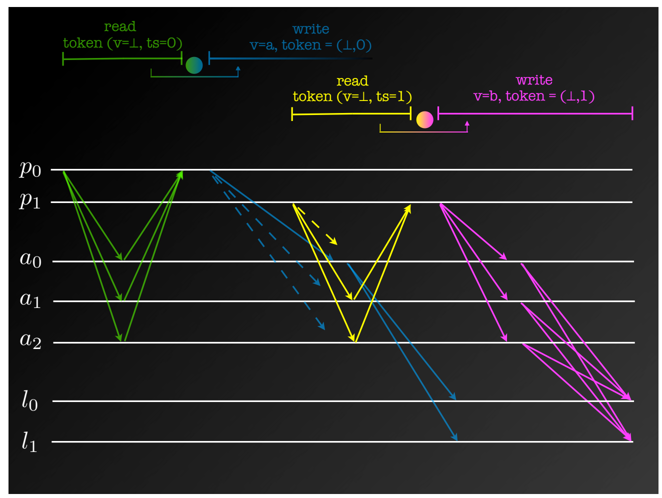


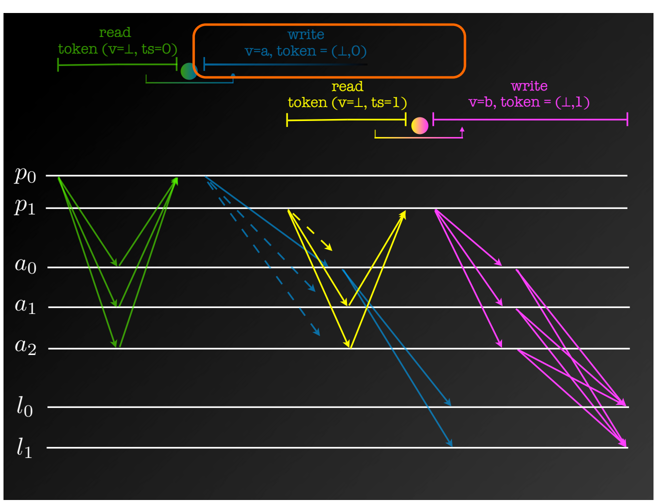
23

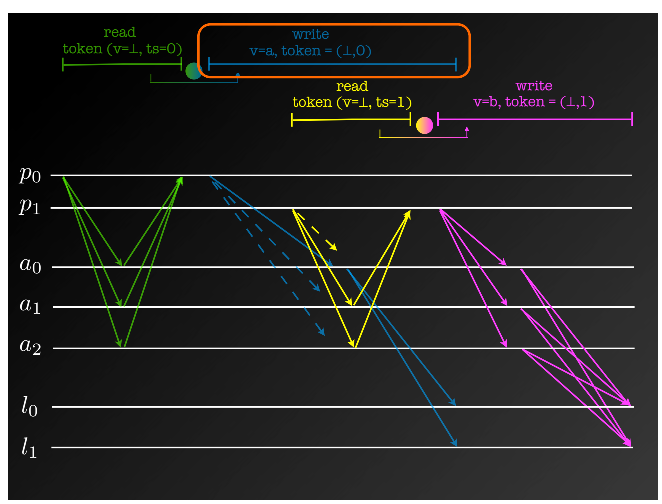

read
token (v=⊥, ts=0)
write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
l_0
l_1

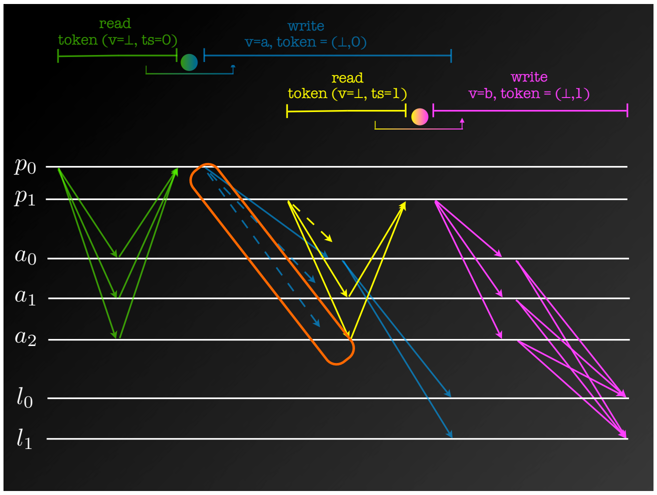

read
token (v=⊥, ts=0)
write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
l_0
l_1


read
token (v=⊥, ts=0)
write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
l_0
l_1

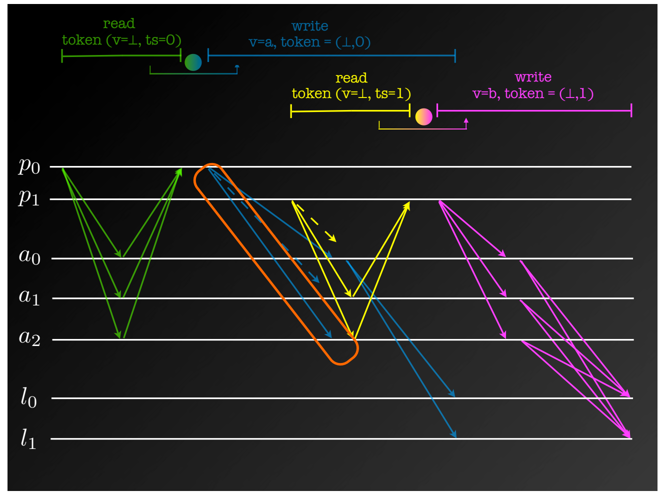

read
token (v=⊥, ts=0)
write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
l_0
l_1


read
token (v=⊥, ts=0)
write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
l_0
l_1

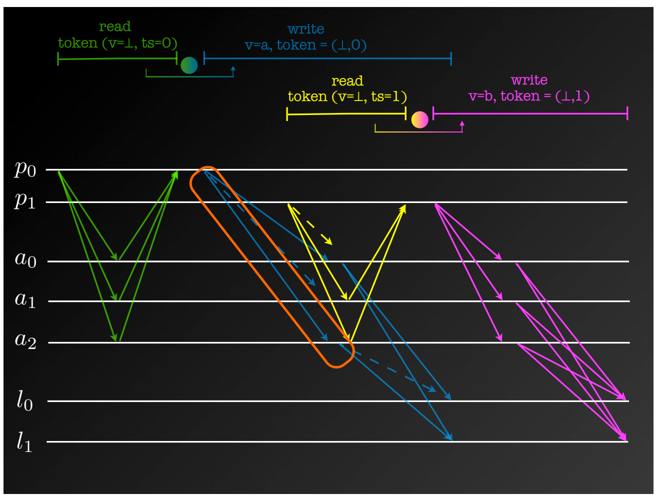

read
token (v=⊥, ts=0)
write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
l_0
l_1



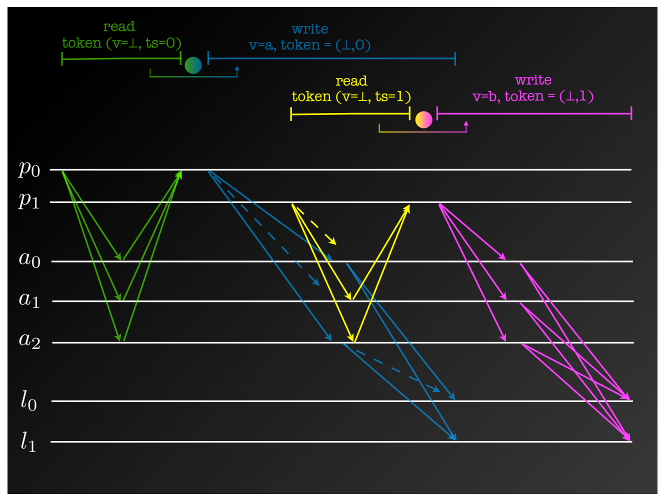
23



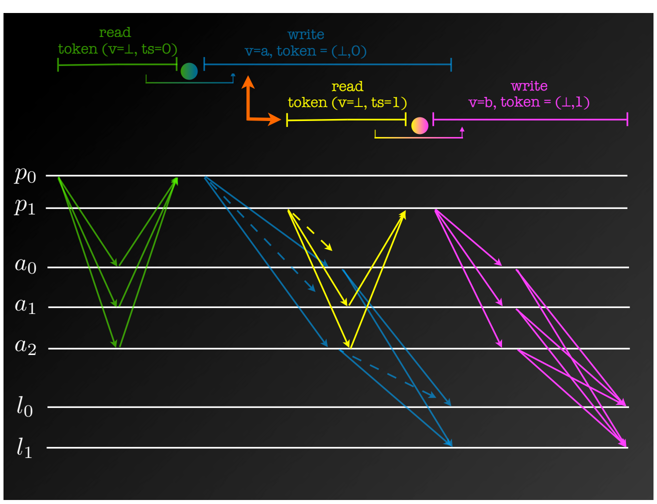
23



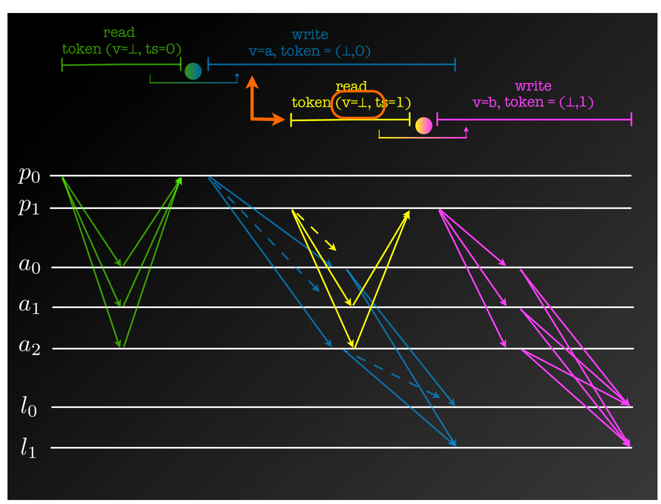
23



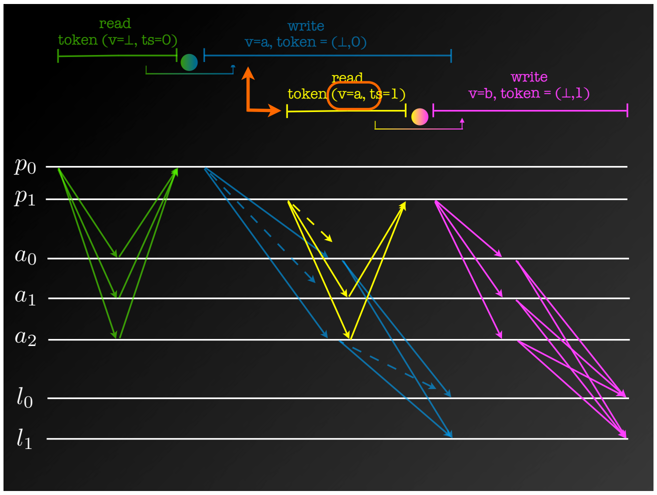
24



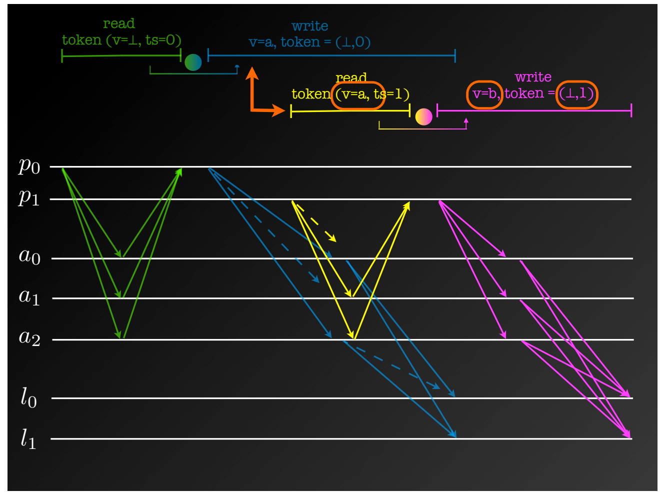
24



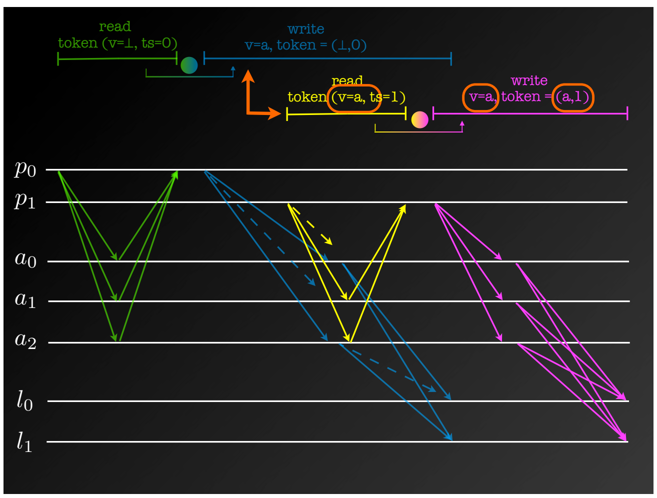
24



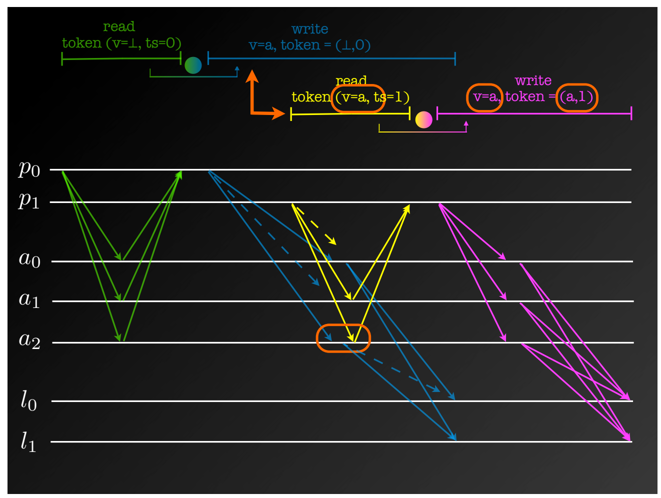
24



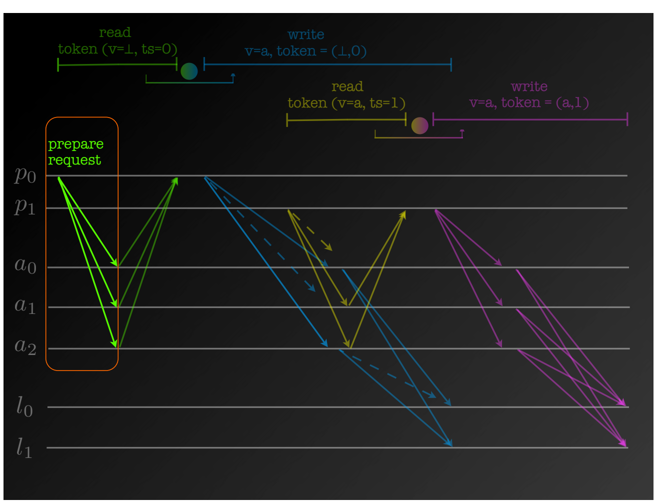
24



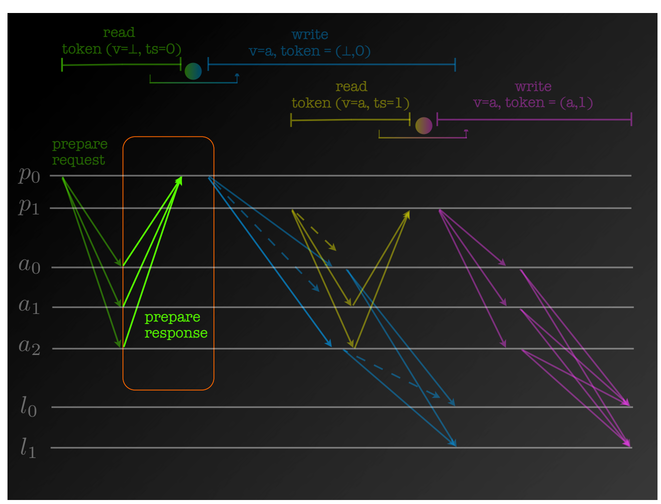
24



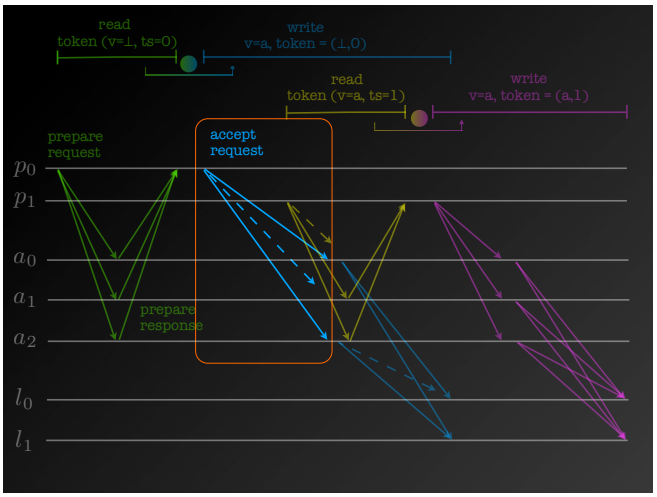
24

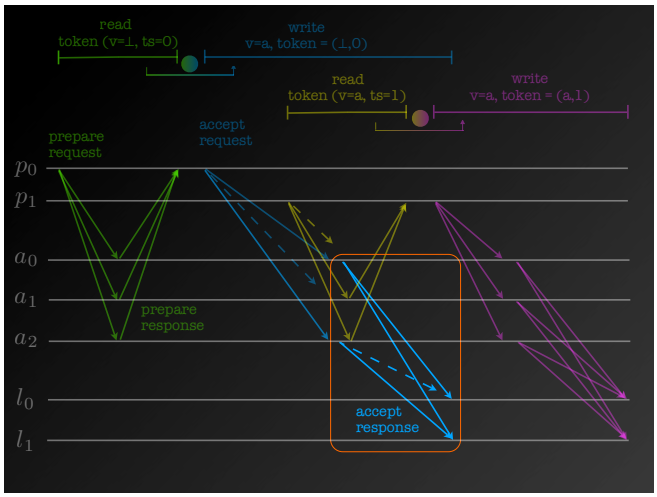


25



26



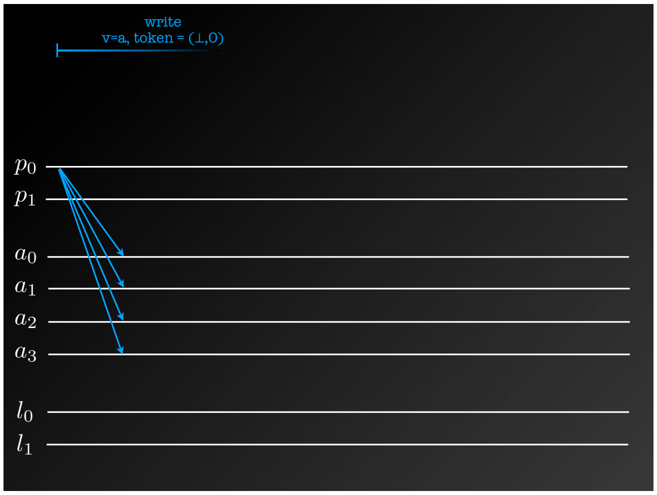


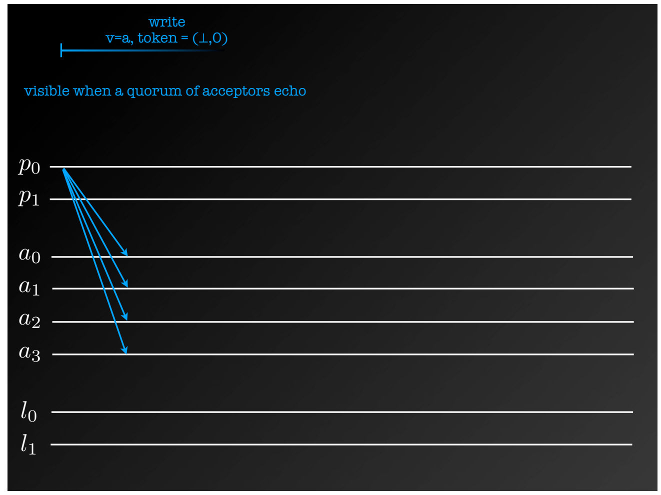
Outline

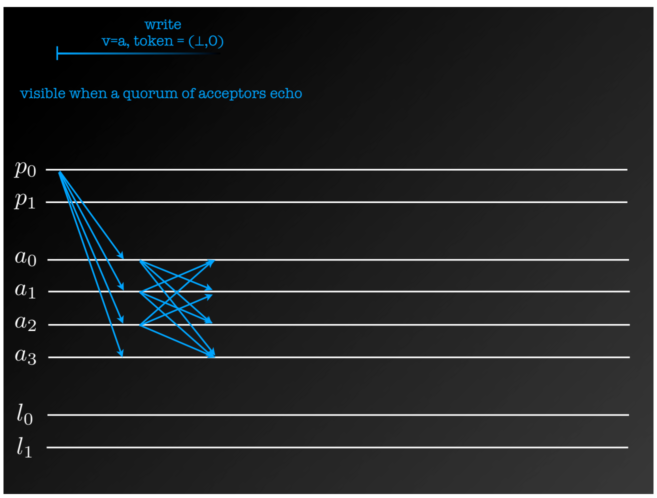
- Motivation
- Paxos Register specification
- Crash implementation
- Byzantine implementation

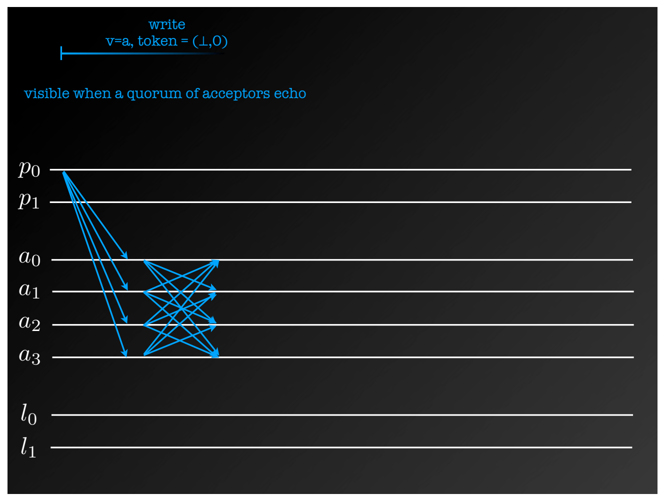
Byzantine Model

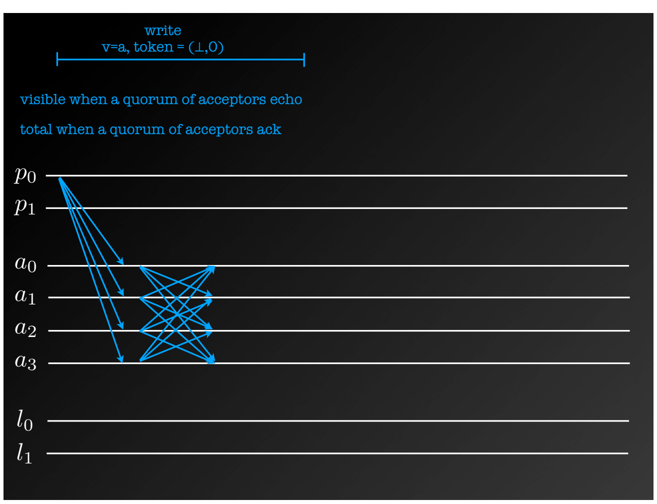
- Asynchronous system
- Digital signatures
- Proposers, acceptors, learners
- At least one proposer correct
- More than two thirds acceptors correct

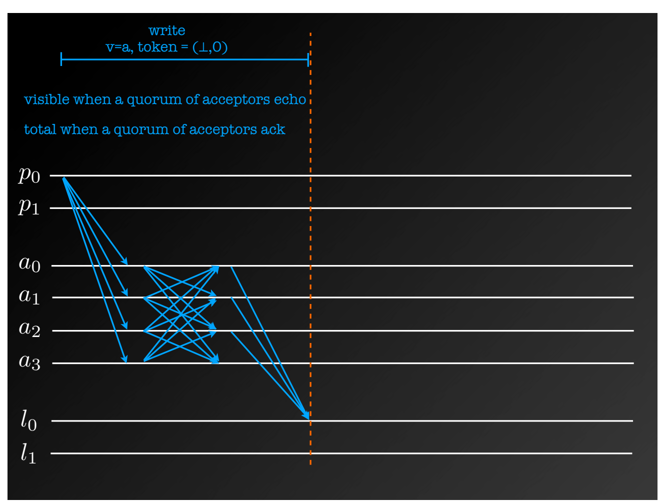

write
v=a, token = (⊥,0)
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1

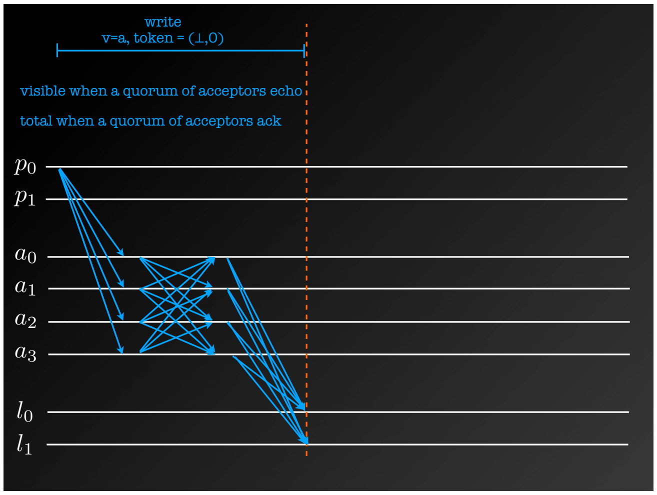

write
v=a, token = (⊥,0)
visible when a quorum of acceptors echo
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1


write
v=a, token = (⊥,0)
visible when a quorum of acceptors echo
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1

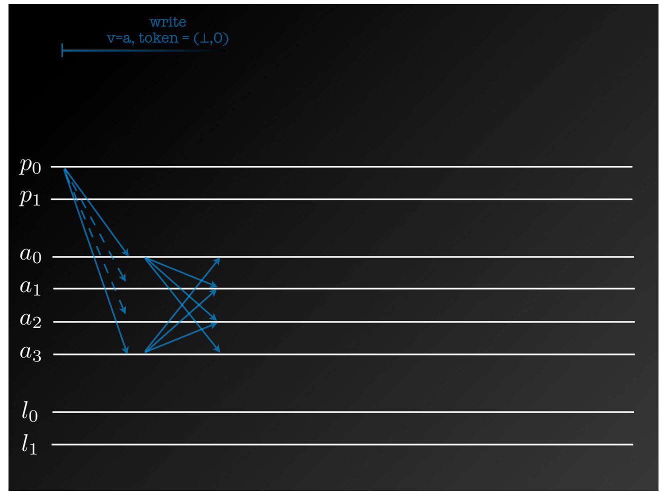

write
v=a, token = (⊥,0)
visible when a quorum of acceptors echo
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1


write
v=a, token = (⊥,0)
visible when a quorum of acceptors echo
total when a quorum of acceptors ack
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1

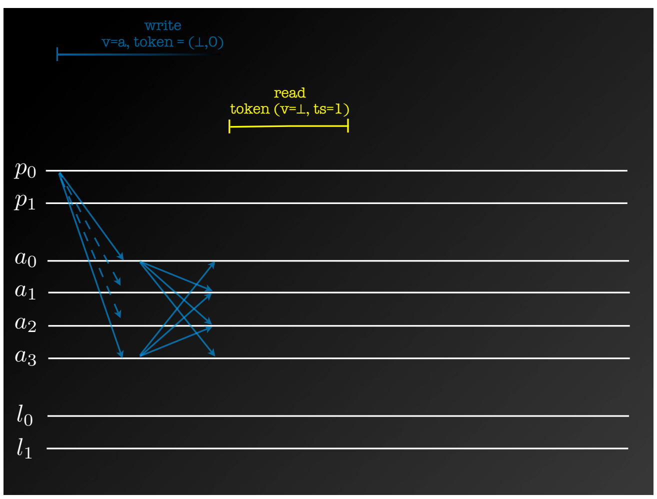

write
v=a, token = (⊥,0)
visible when a quorum of acceptors echo
total when a quorum of acceptors ack
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1



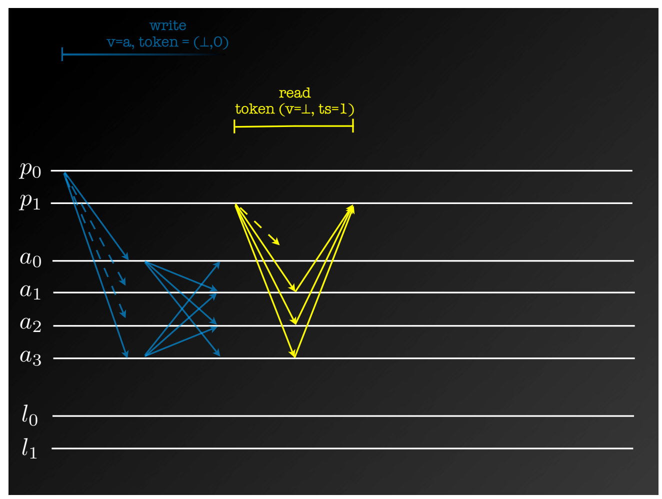
31



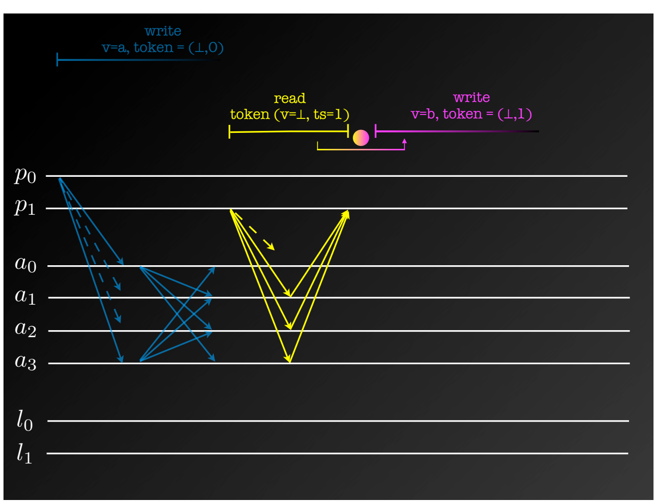
32



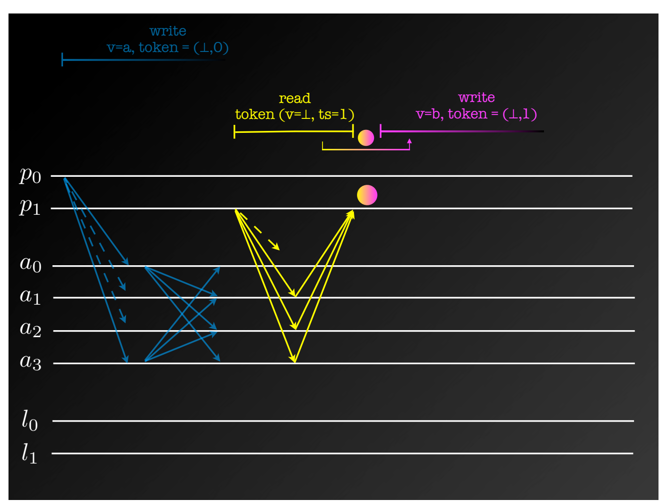
32



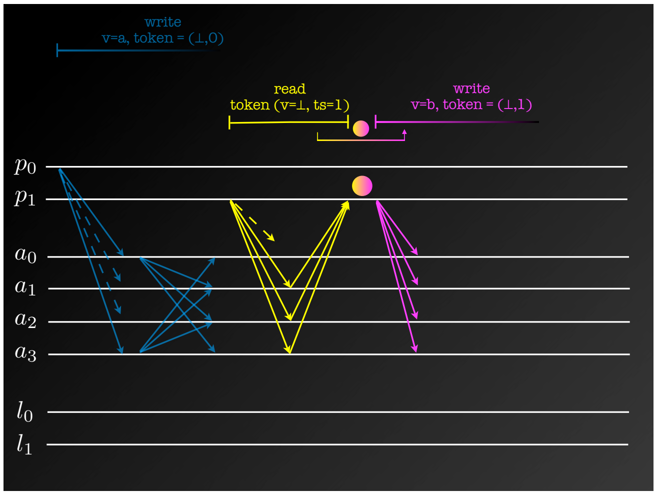
32

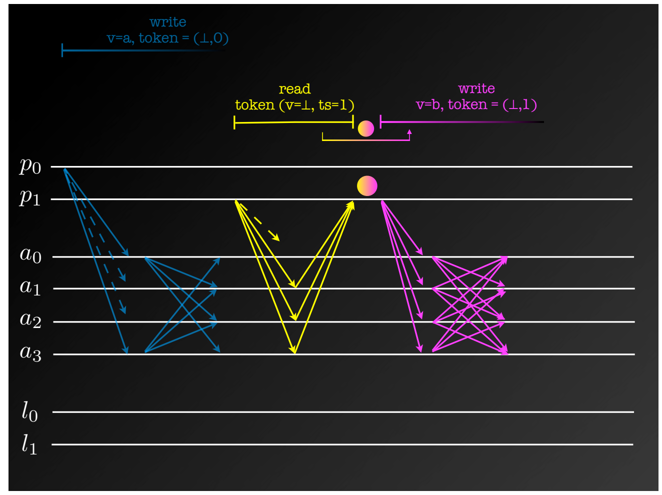


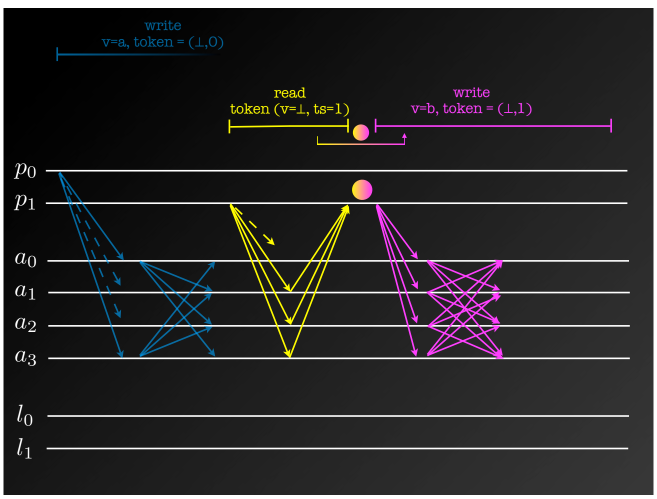
32

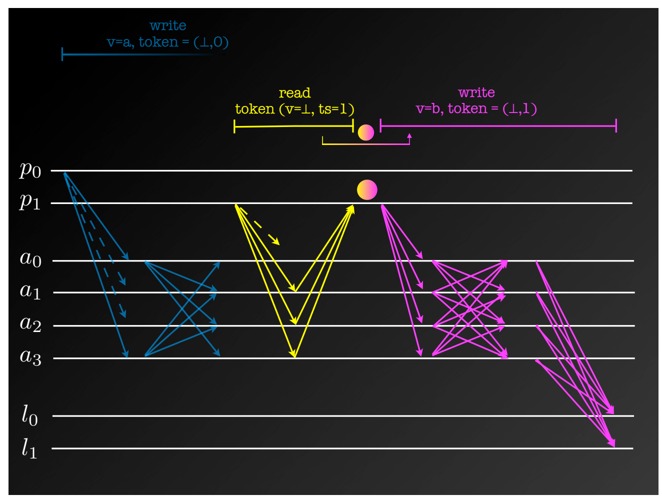


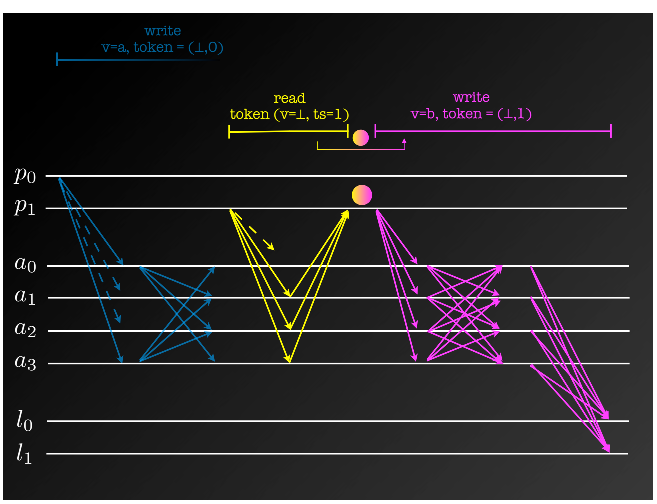
32

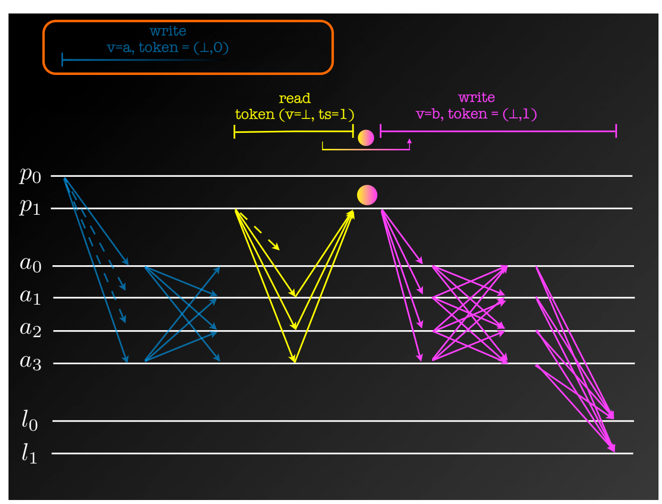

write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1

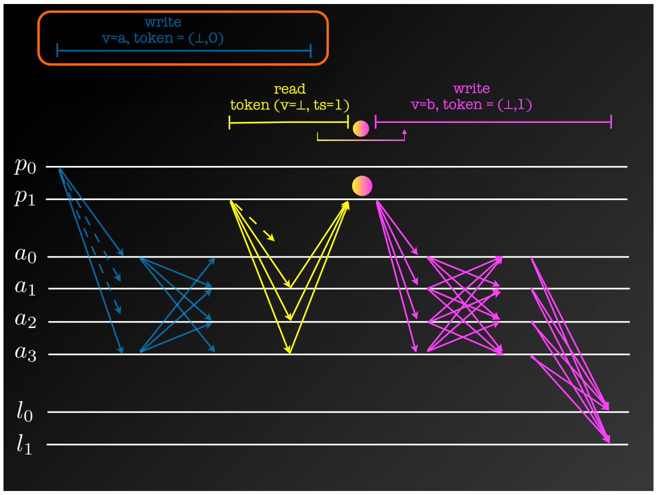

write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1


write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1

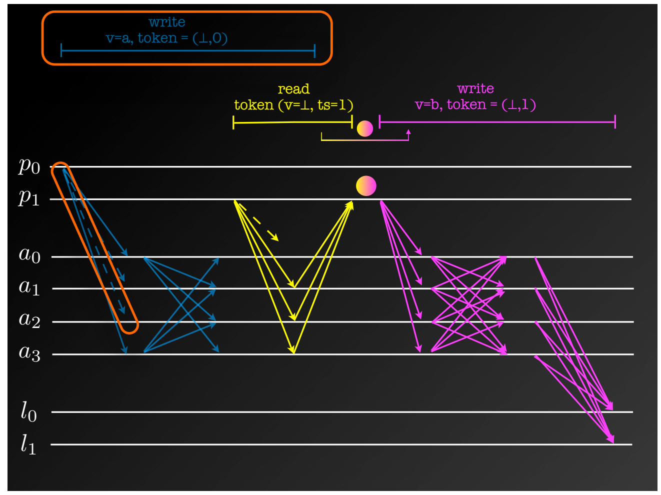

write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1


write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1

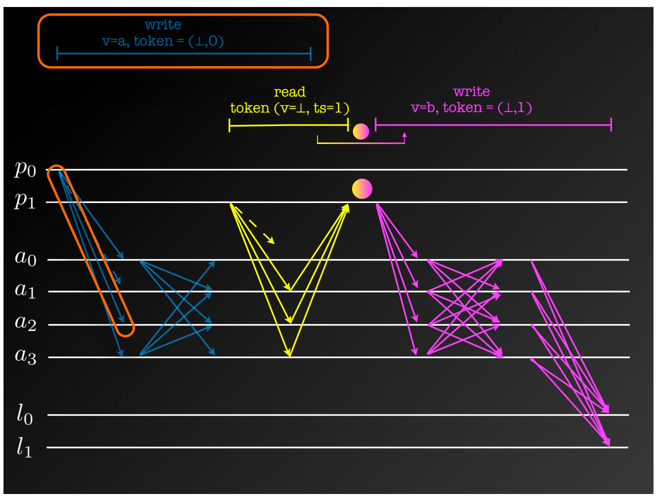

write
v=a, token = (⊥,0)
read
token (v=⊥, ts=1)
write
v=b, token = (⊥,1)
p_0
p_1
a_0
a_1
a_2
a_3
l_0
l_1



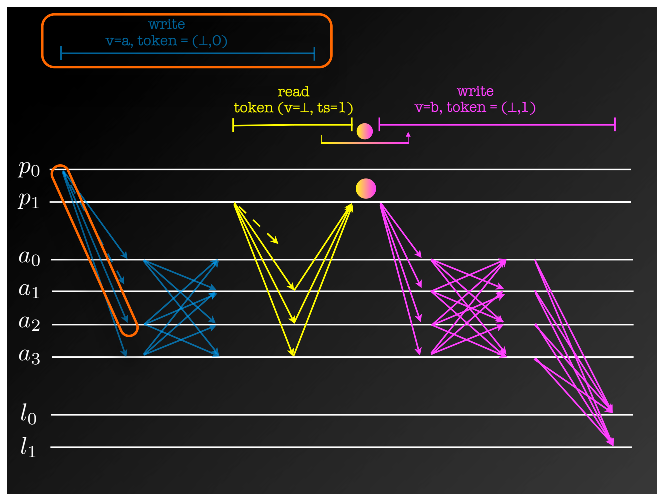
33



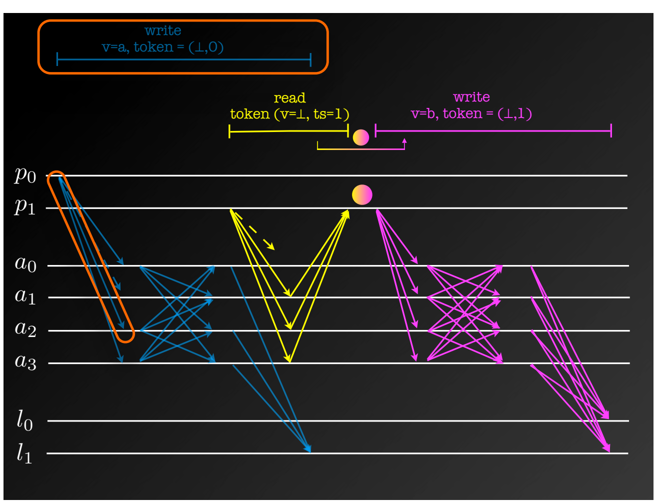
33



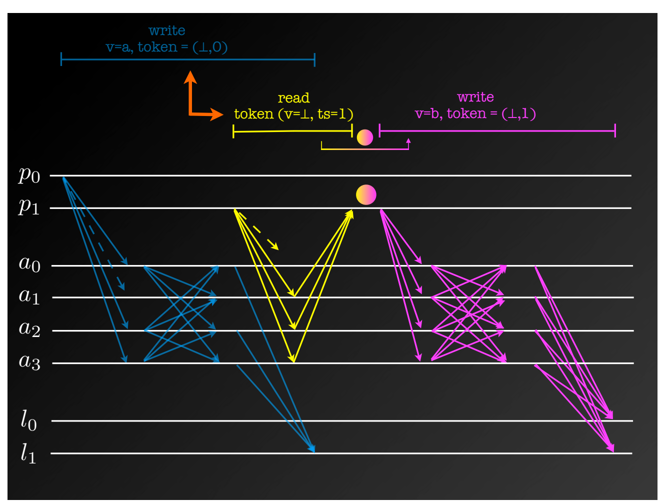
33



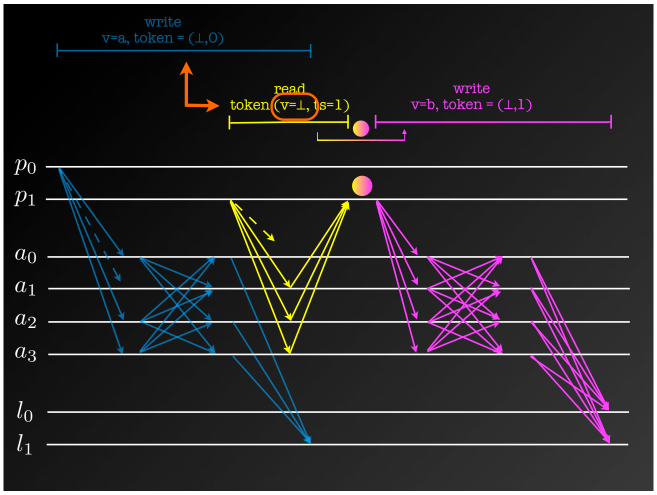
33



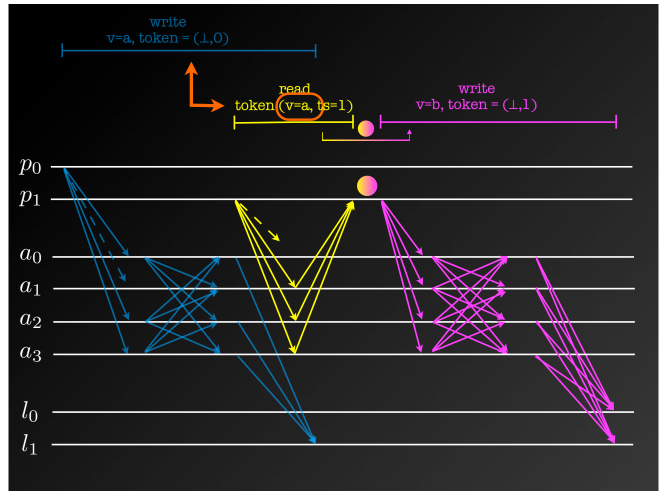
33



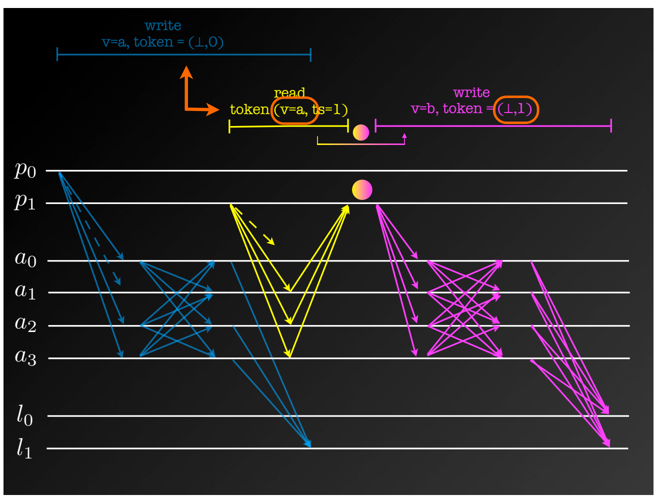
34



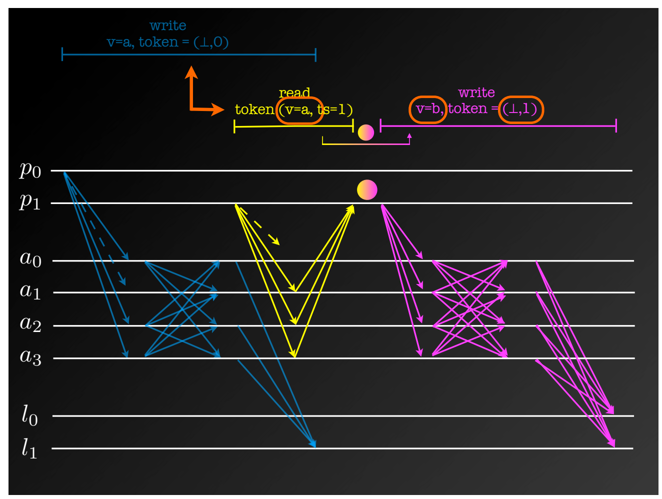
34



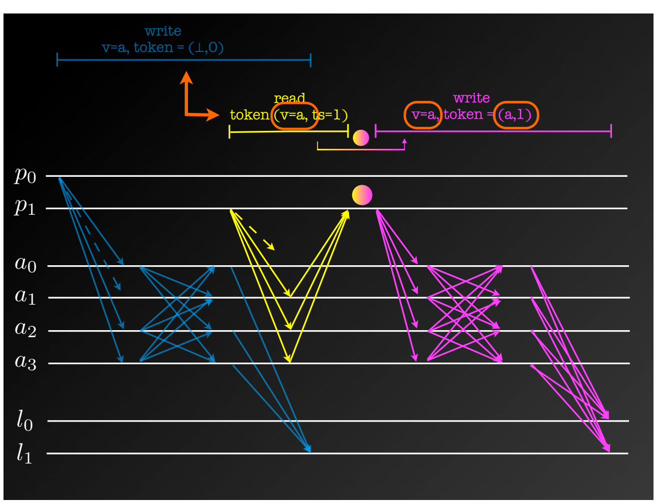
34



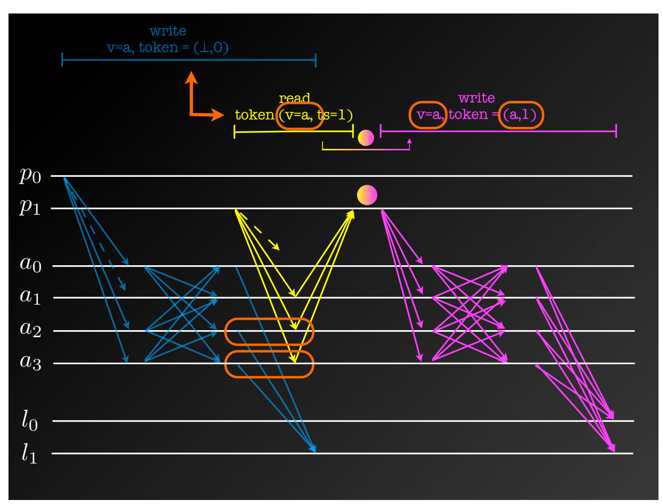
34



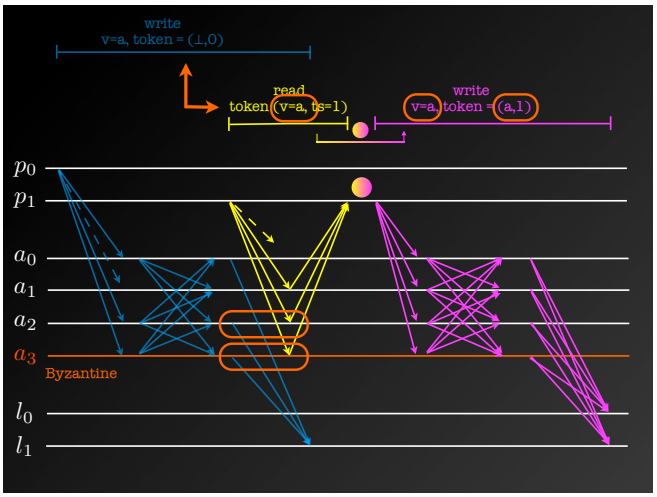
34



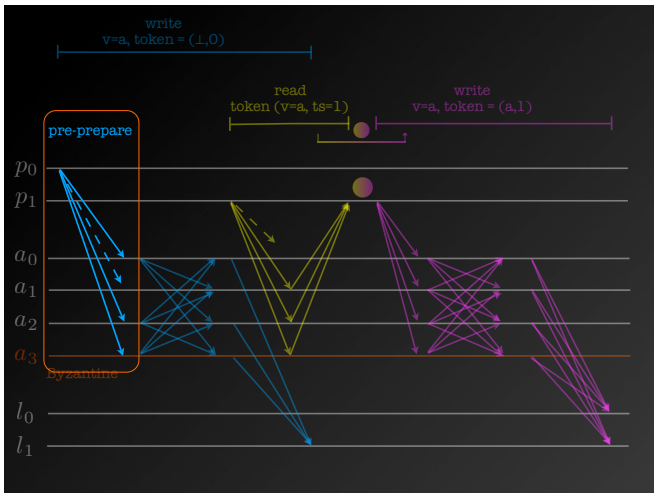
34



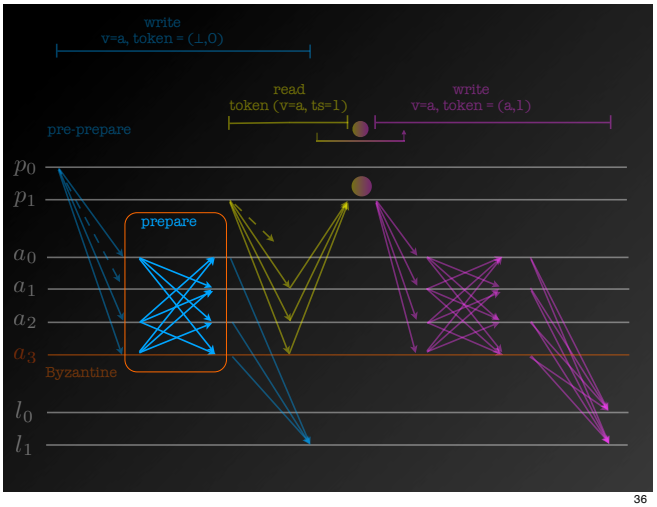
34



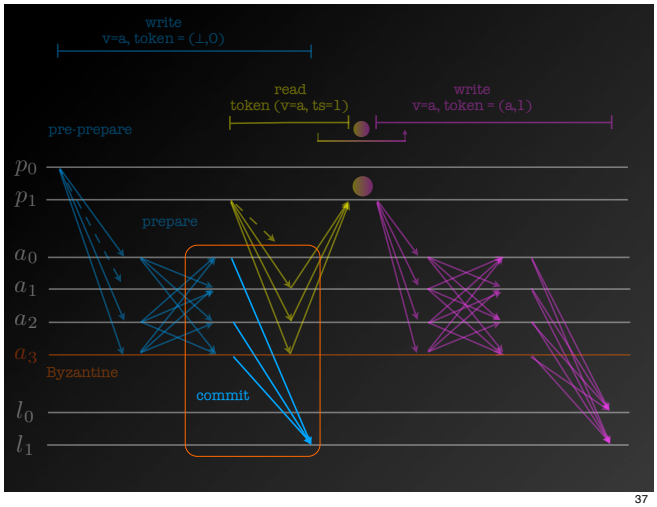
34



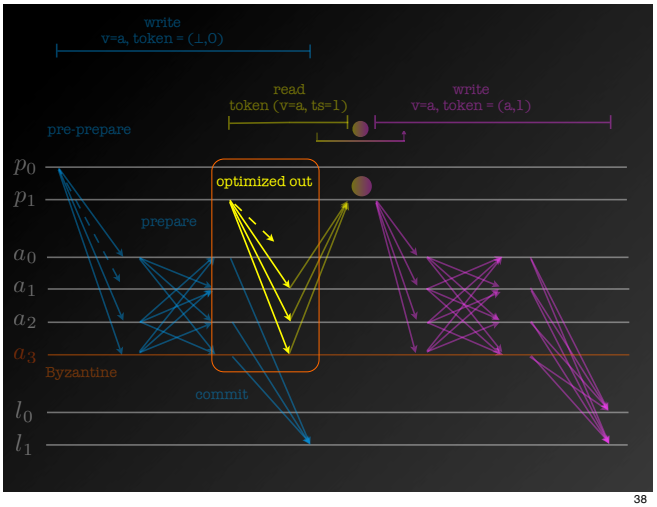
35



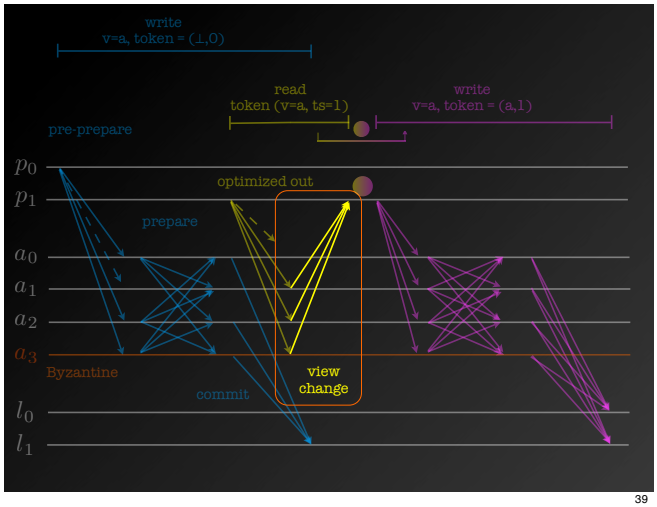
36



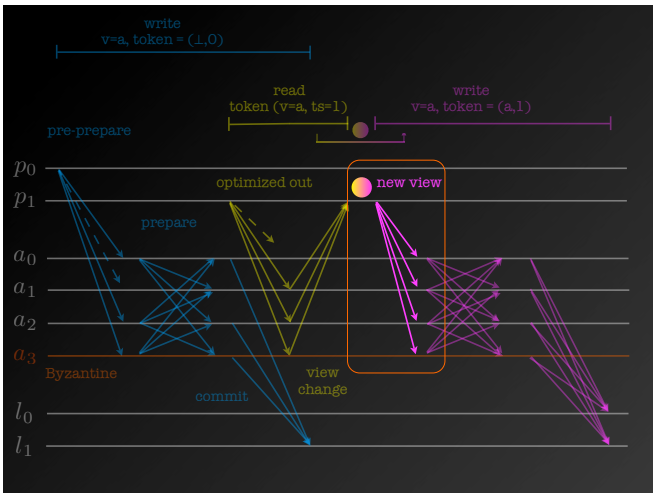
37



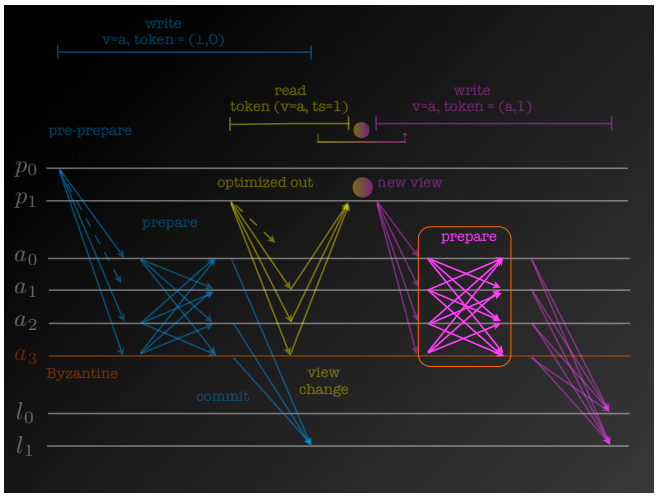
38



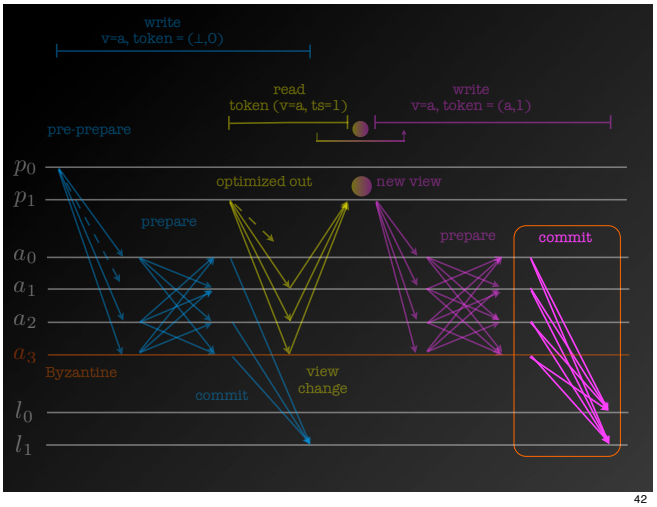
39



40



41



42

Take Away Points

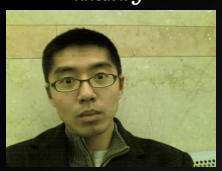
Paxos Register

- Captures similarities in Paxos protocols
- Hides implementation details
- Important abstractions:

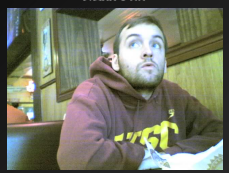
  tokens, visible writes, total writes

43

Harry



Allen



Amit

Lorenzo

44