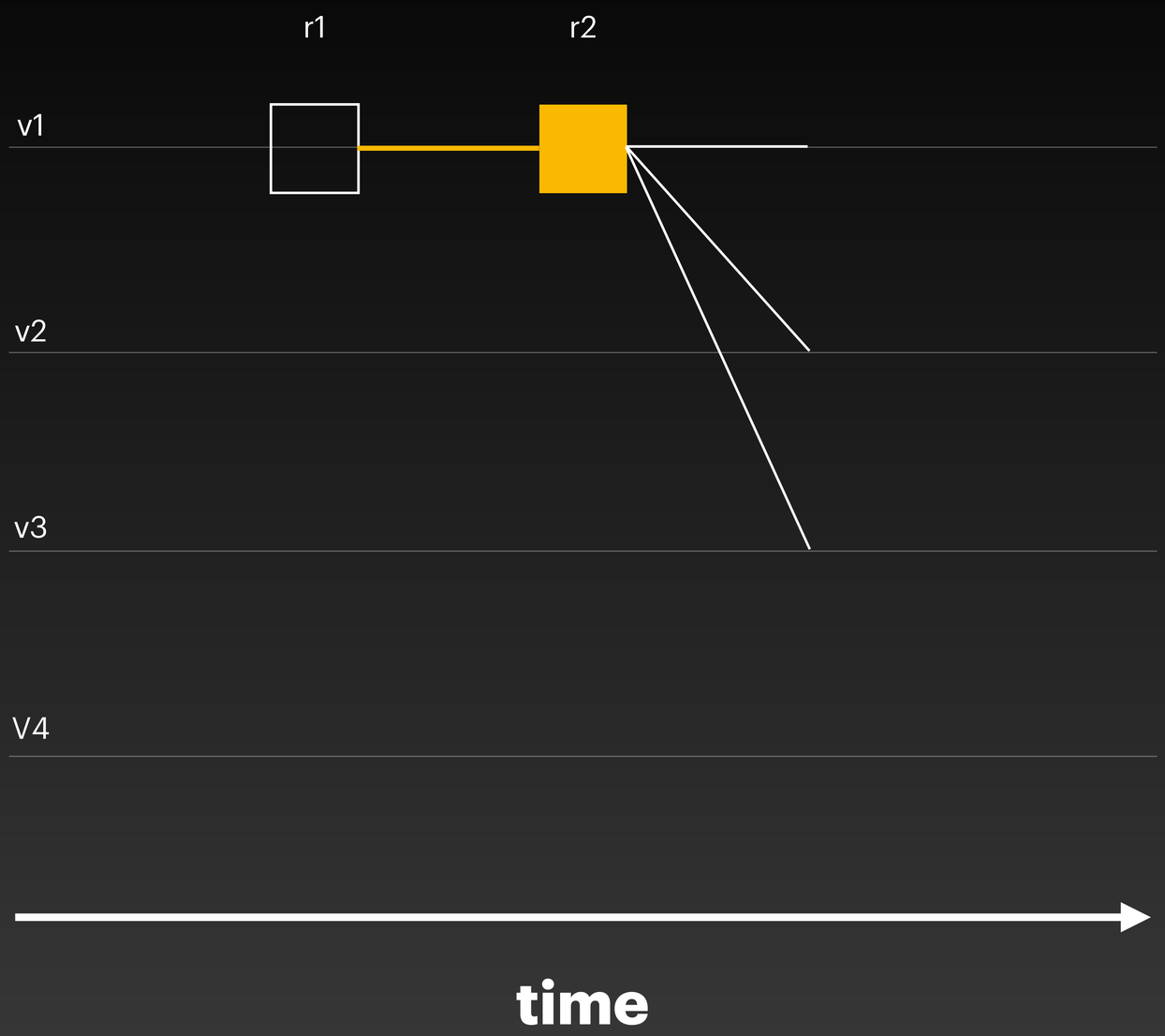
DAG-Based Consensus

Introduction and practical Aspects

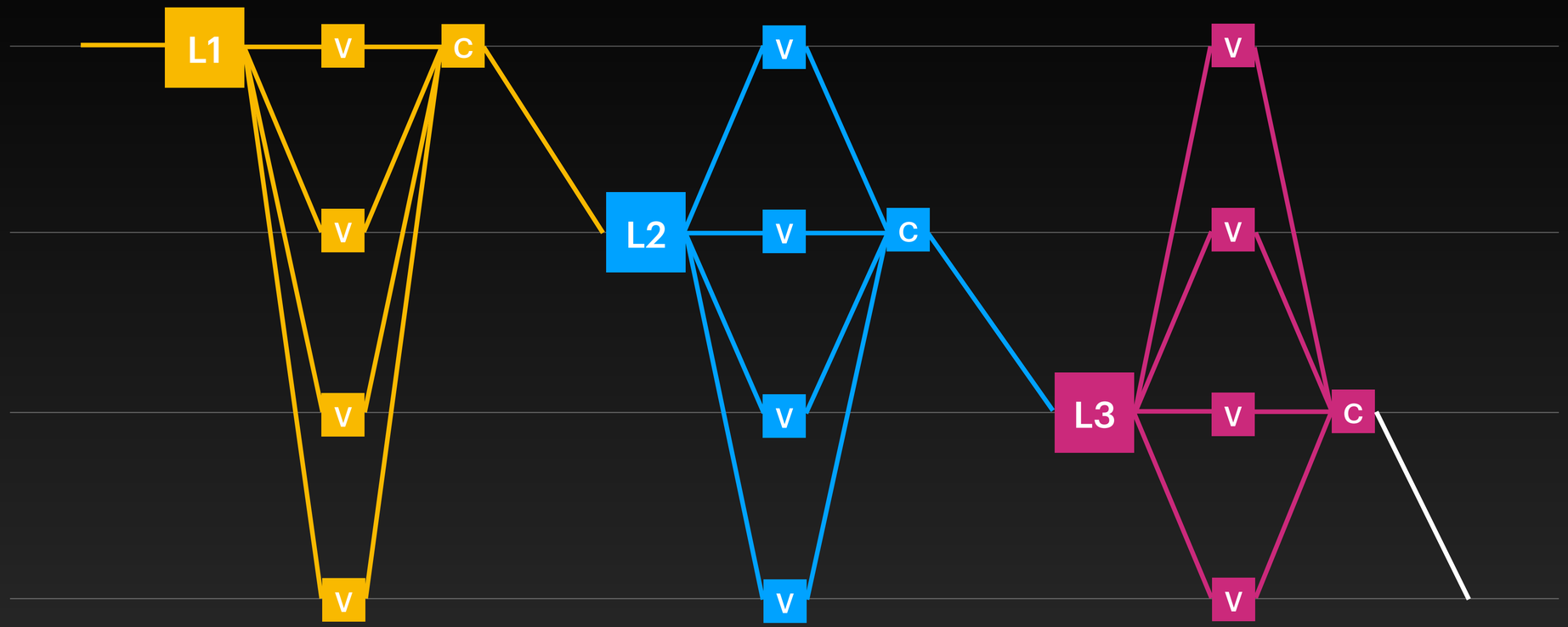
Alberto Sonnino

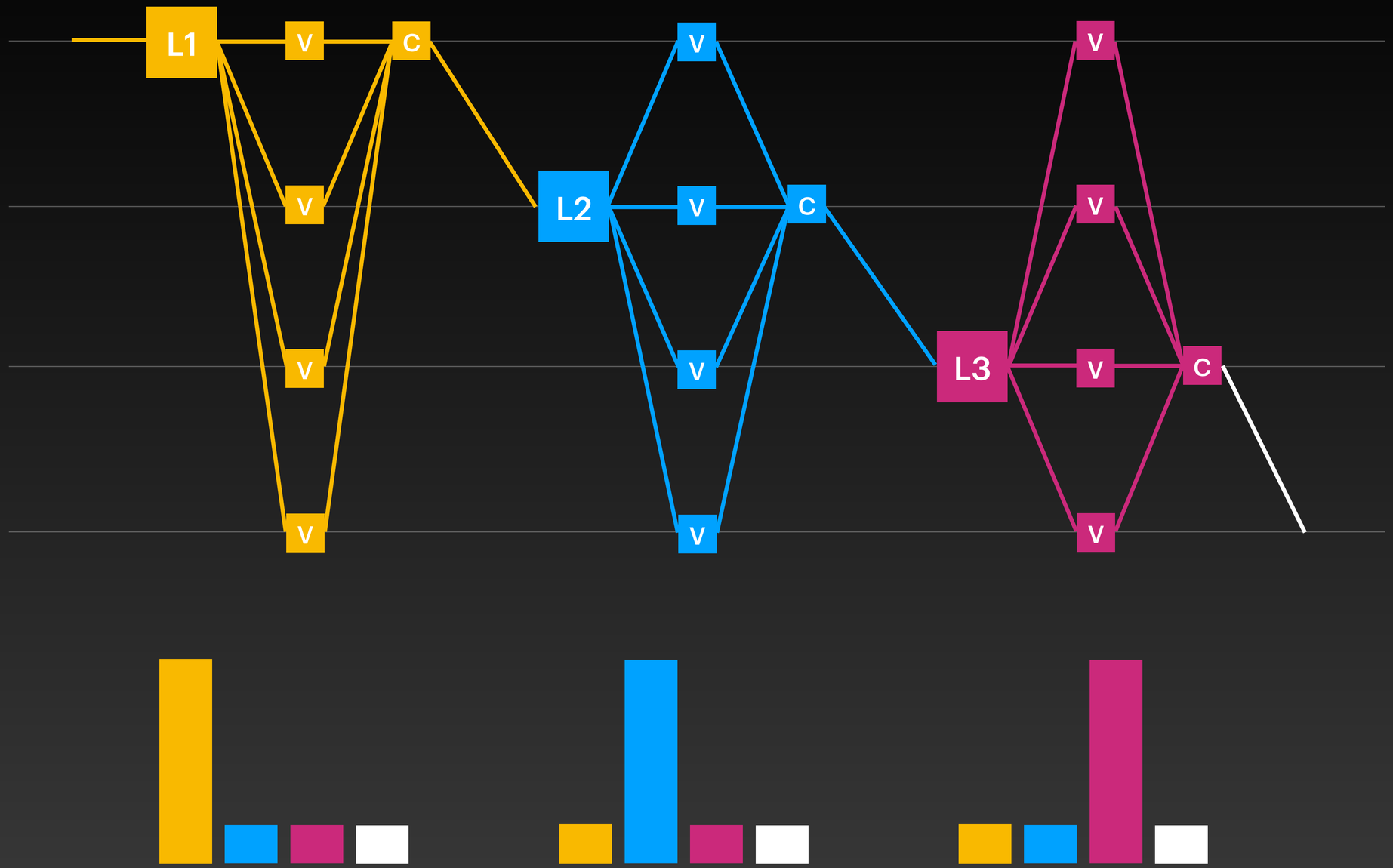
> 2/3

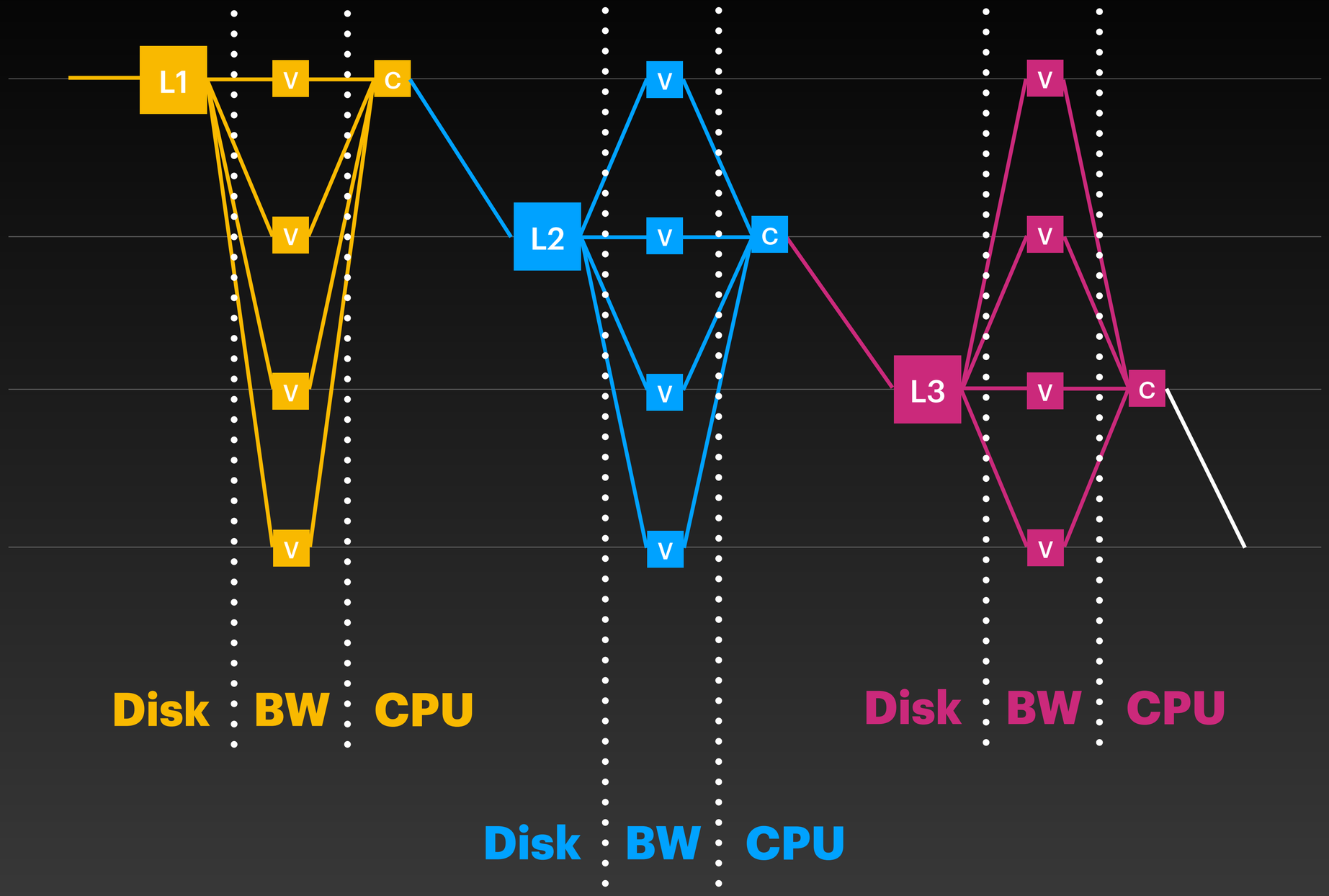
Partial Synchrony

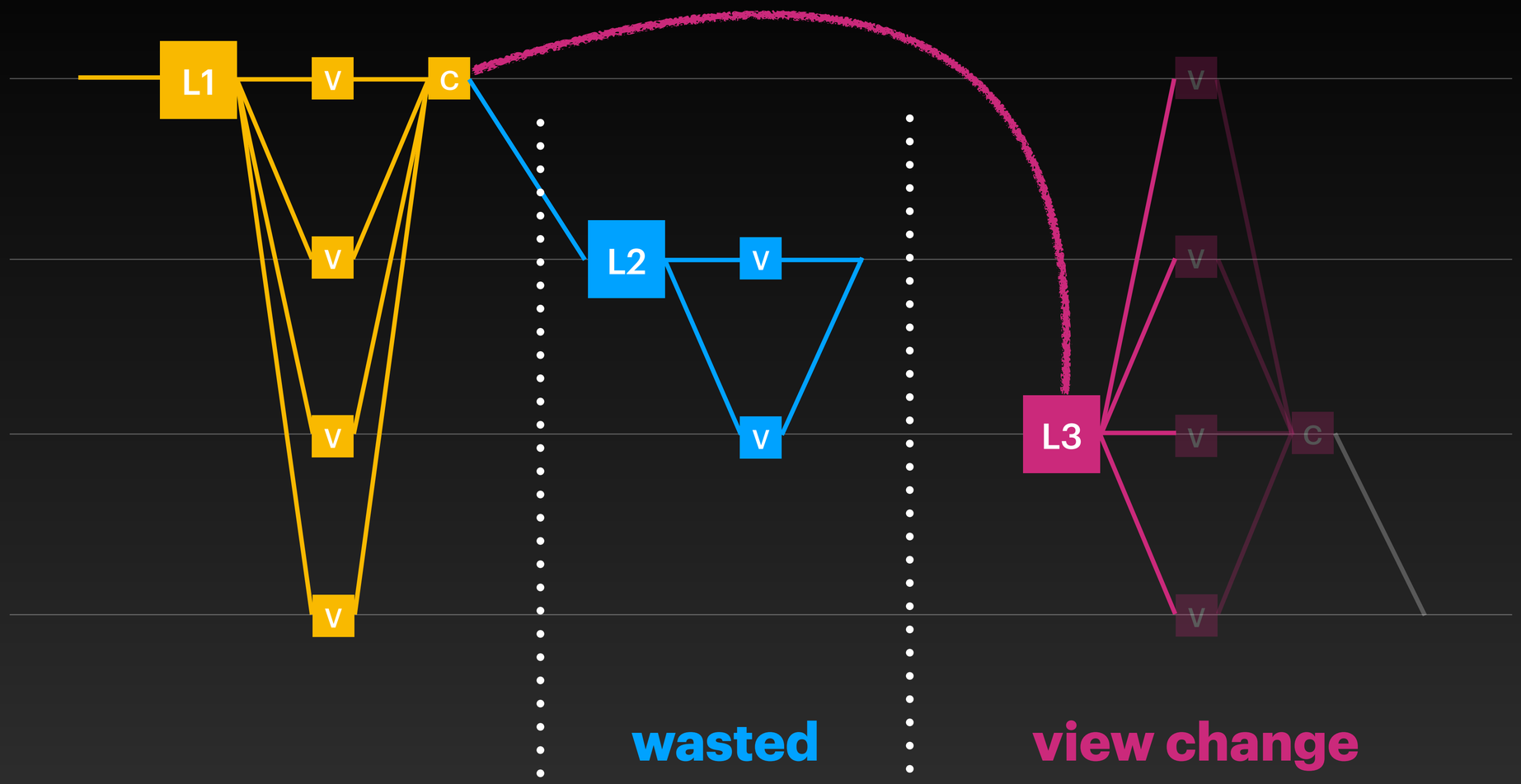


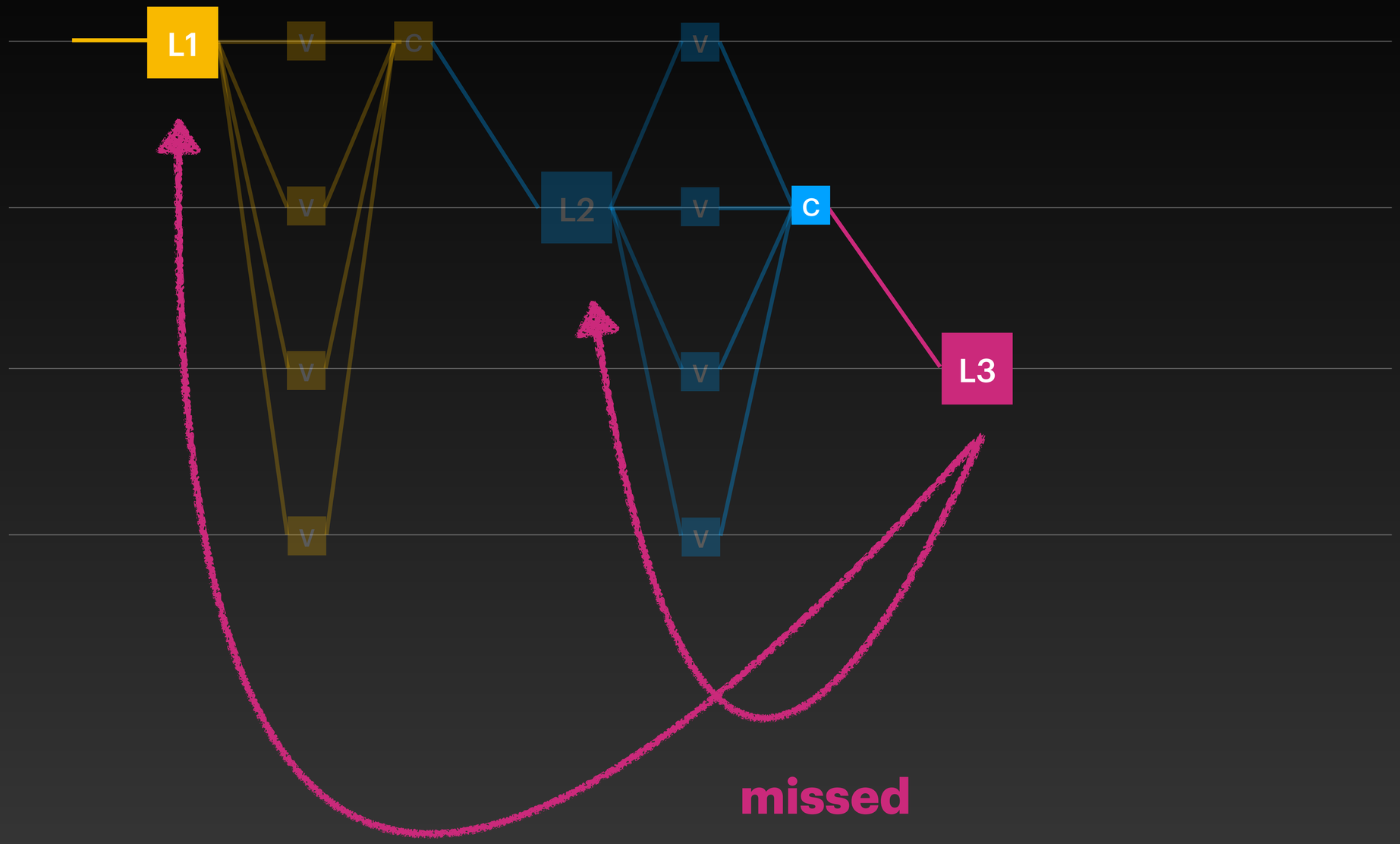
- Round number
- Author
- Payload (txs)
- Parent
- Signature

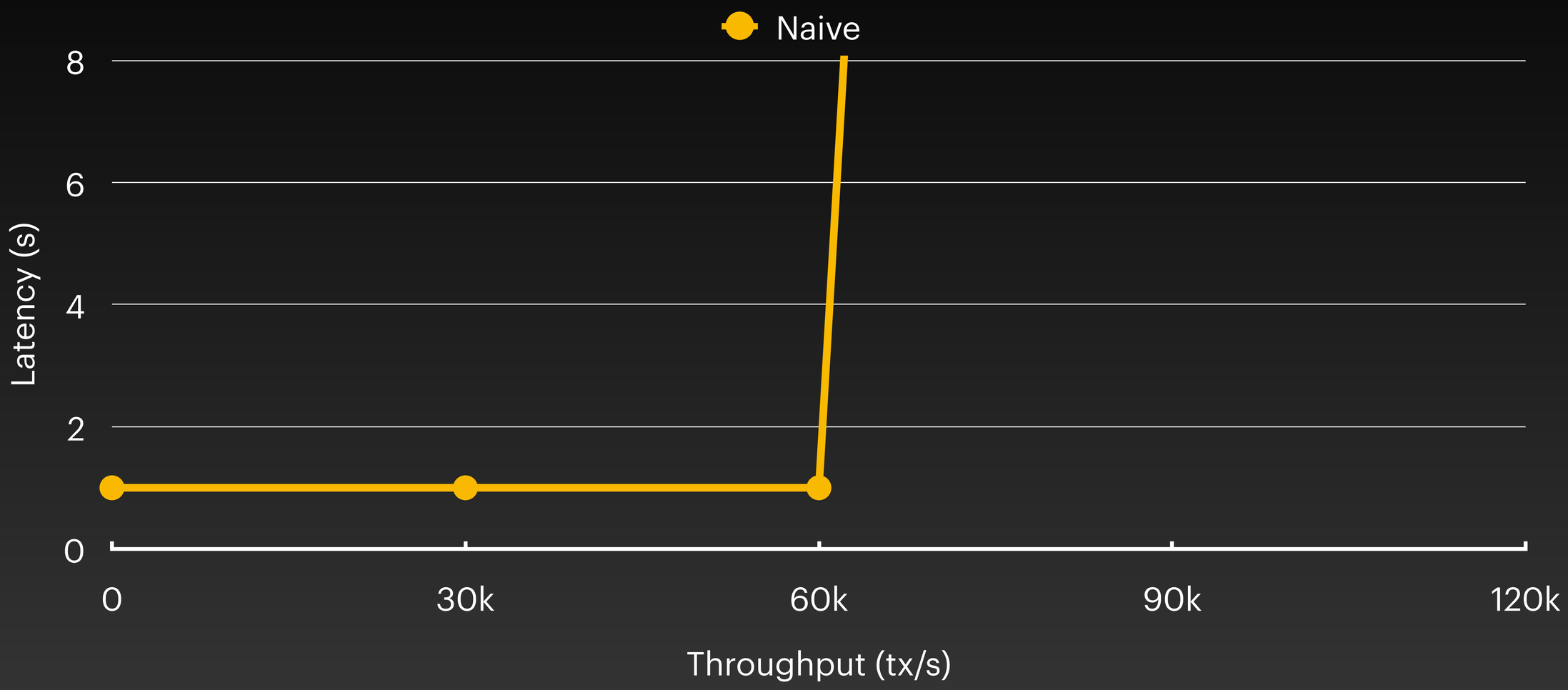

L1
V
V
V
V
C
L2
V
V
V
V
C
L3
V
V
V
V
C

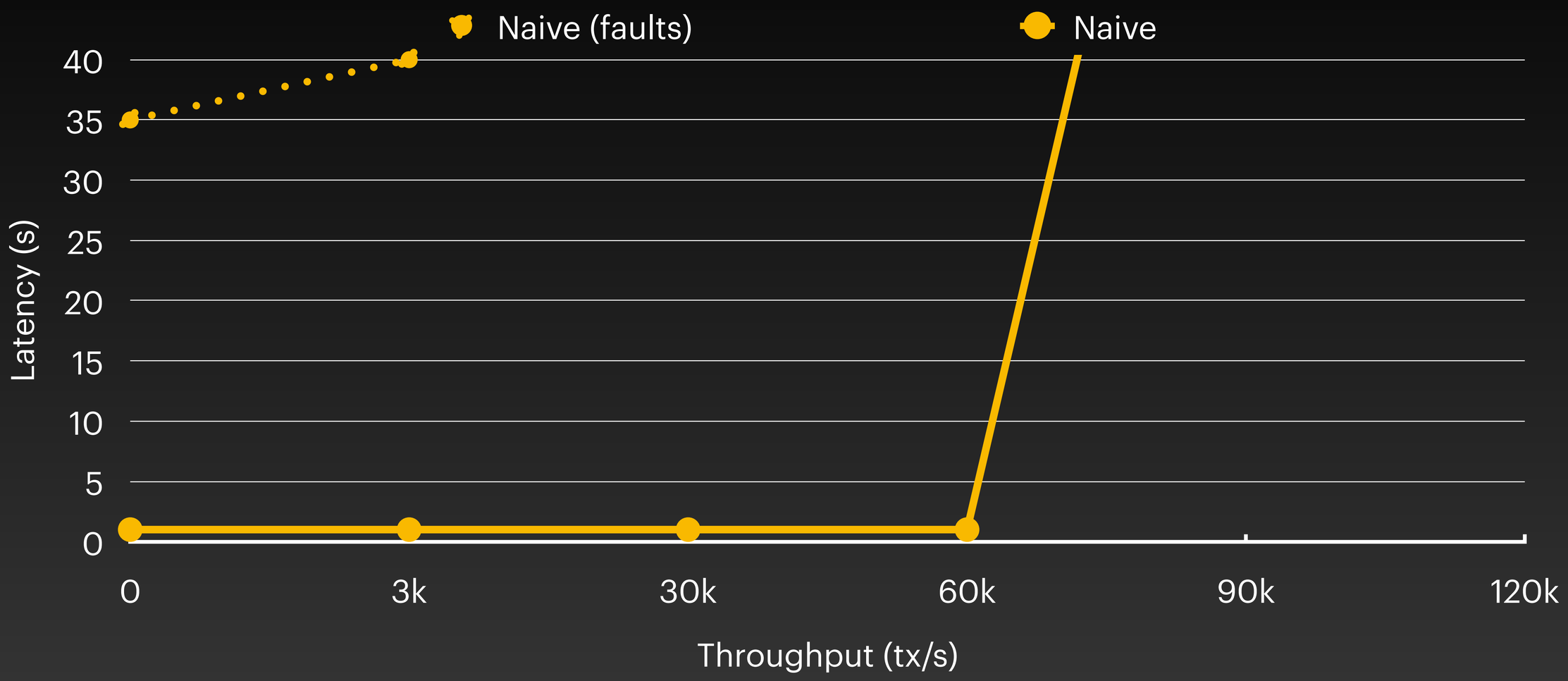

L1
V
V
V
V
C
L2
V
V
V
V
C
L3
V
V
V
V
C

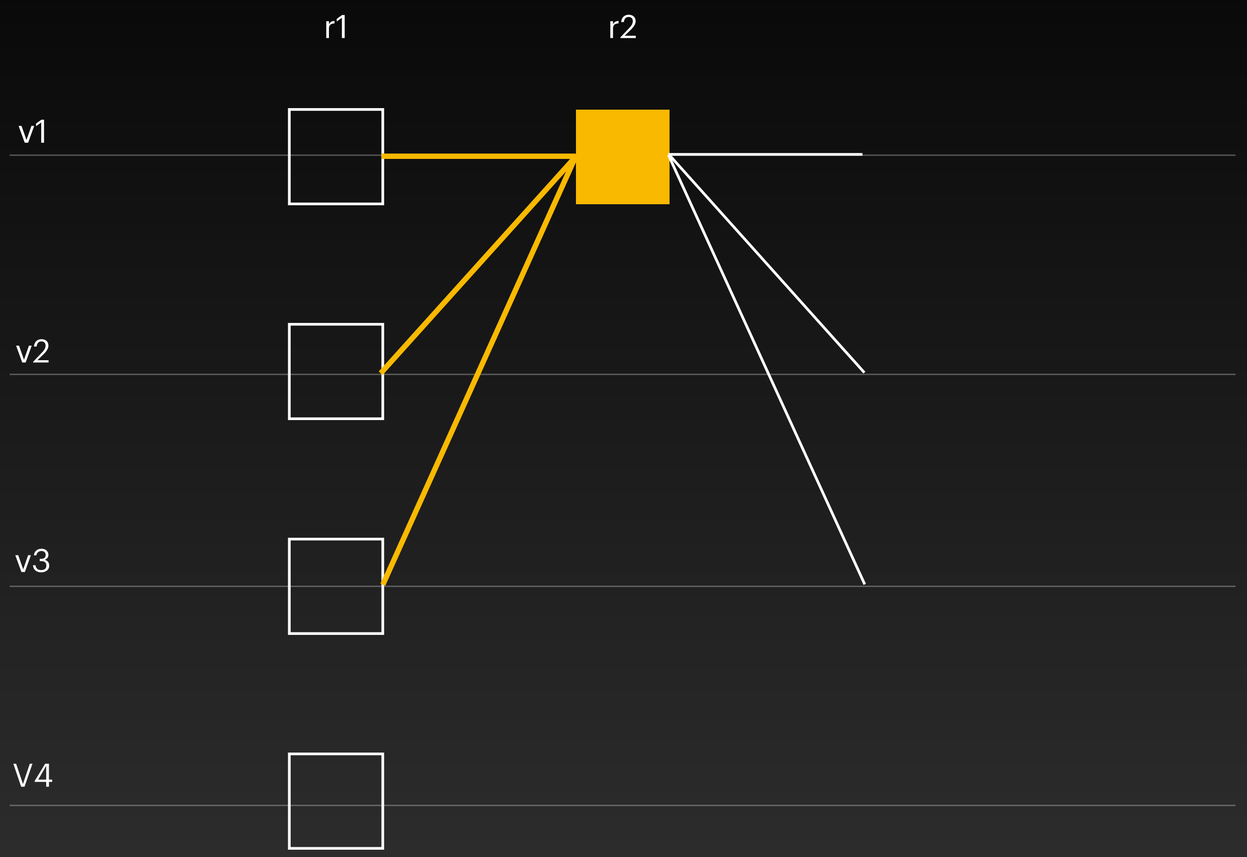

L1
V
V
V
V
C
L2
V
V
V
C
L3
V
V
V
C
Disk
BW
CPU
Disk
BW
CPU
Disk
BW
CPU


L1
V
C
V
V
V
L2
V
V
L3
V
V
V
V
C
wasted
view change


L1
V
C
V
V
V
V
L2
V
V
V
C
L3
missed

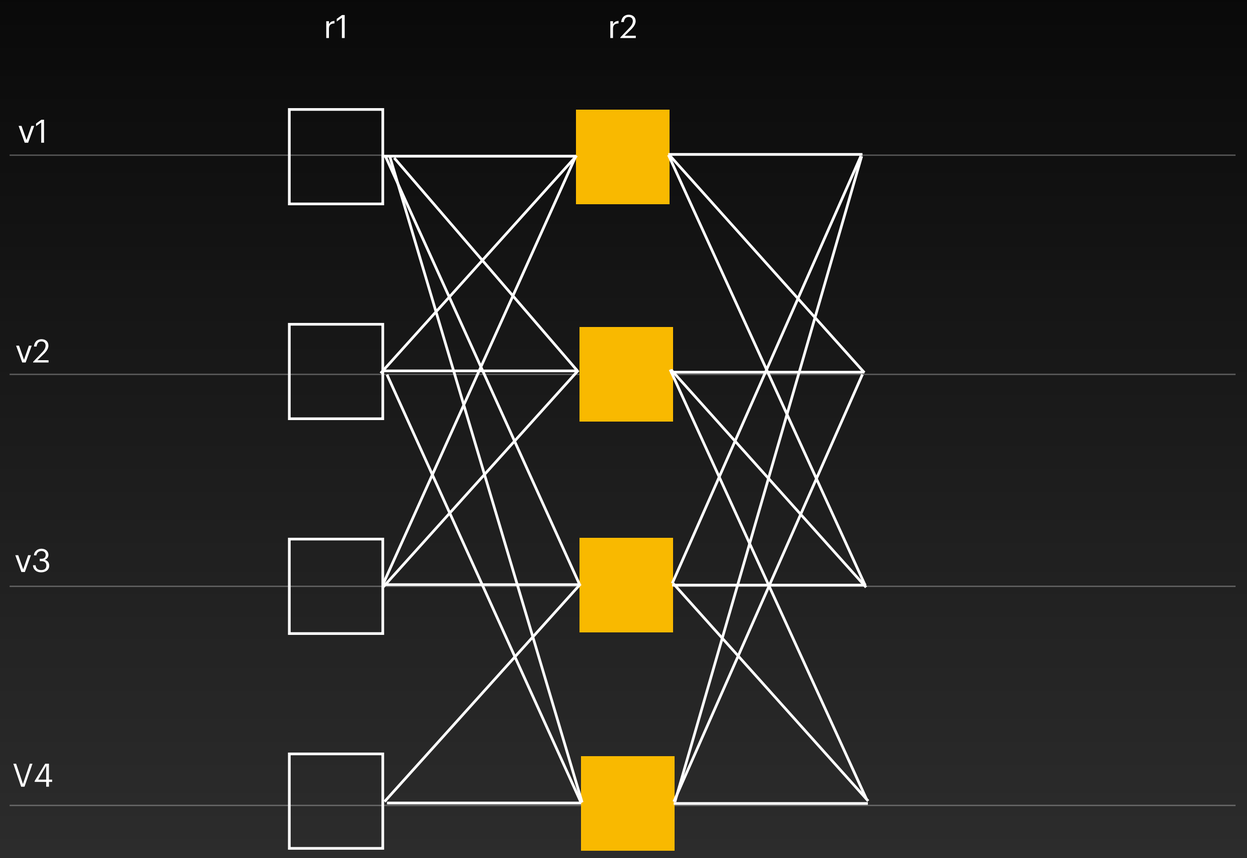

Naive
8
6
4
2
0
Latency (s)
0
30k
60k
90k
120k
Throughput (tx/s)


Naive (faults)
Naive
40
35
30
25
20
15
10
5
0
Latency (s)
0
3k
30k
60k
90k
120k
Throughput (tx/s)



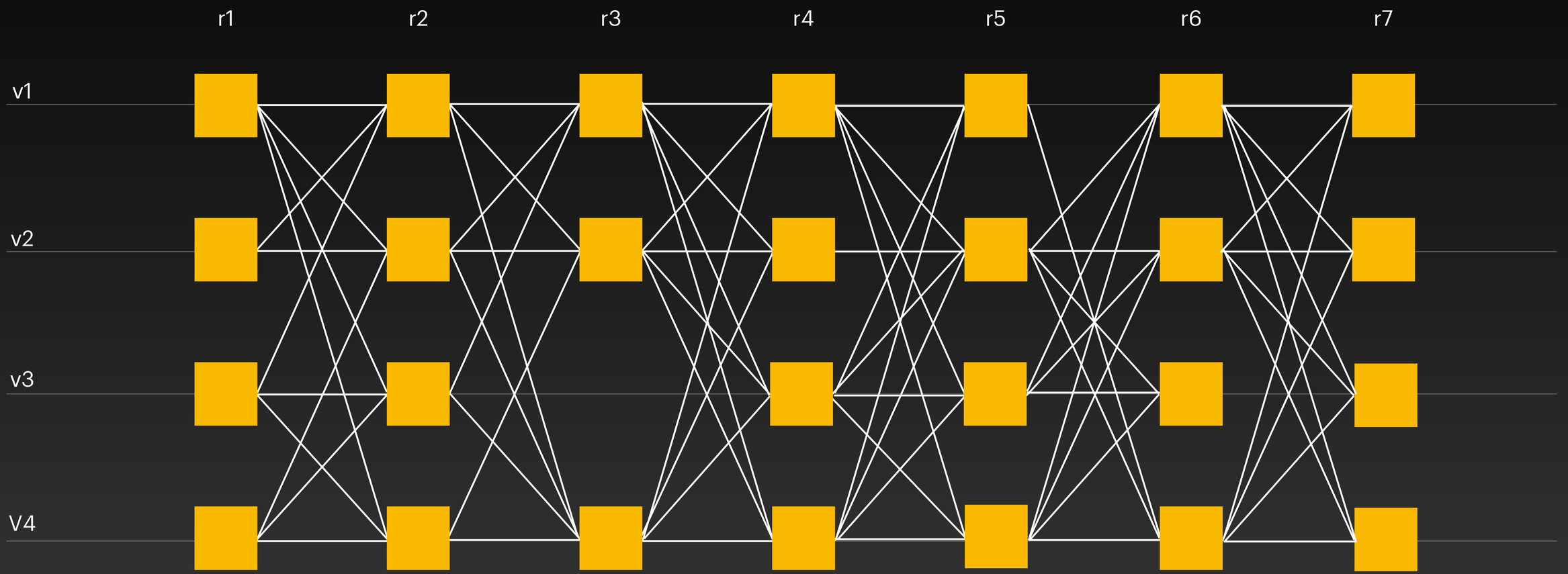
- Round number
- Author
- Payload (txs)
- **Parents**
- Signature

Back Pressure



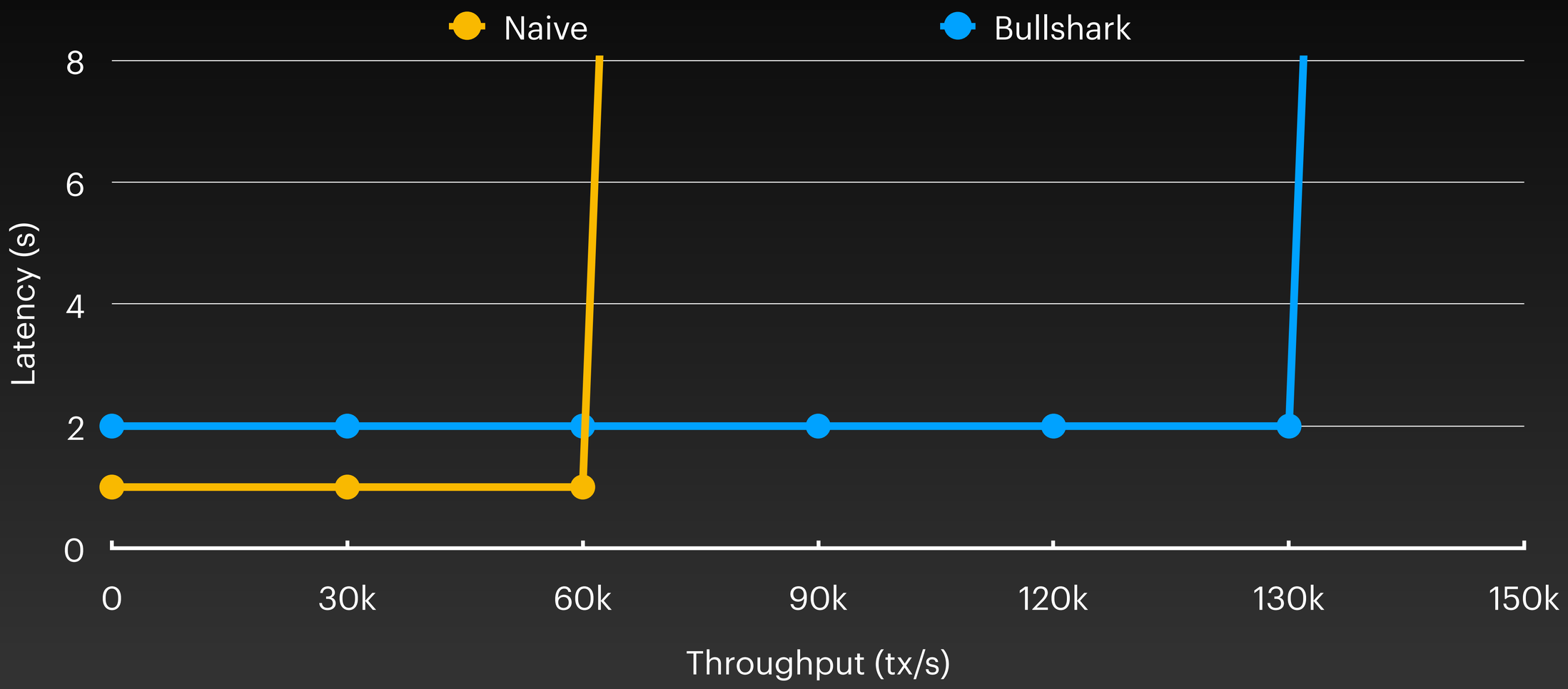
- Round number
- Author
- Payload (txs)
- Parents
- Signature

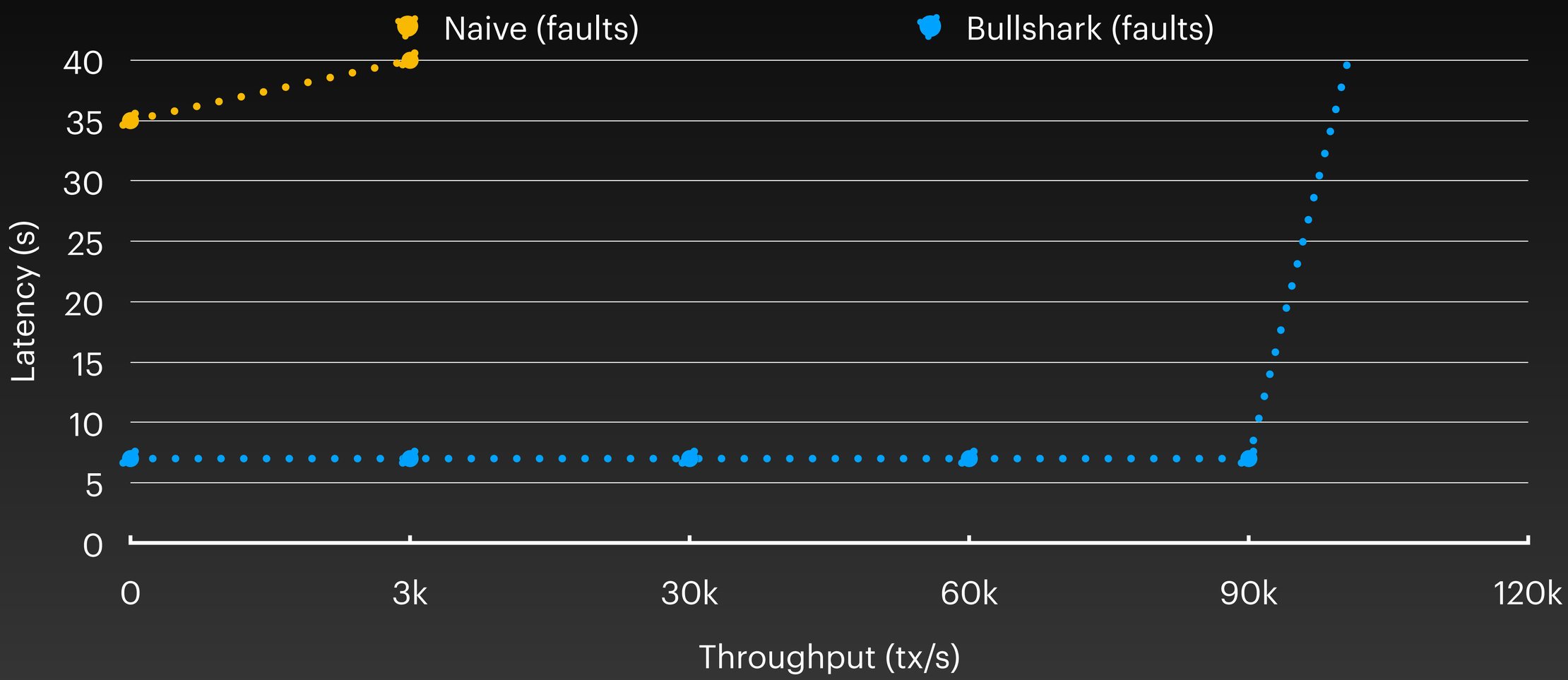
Multi Proposer

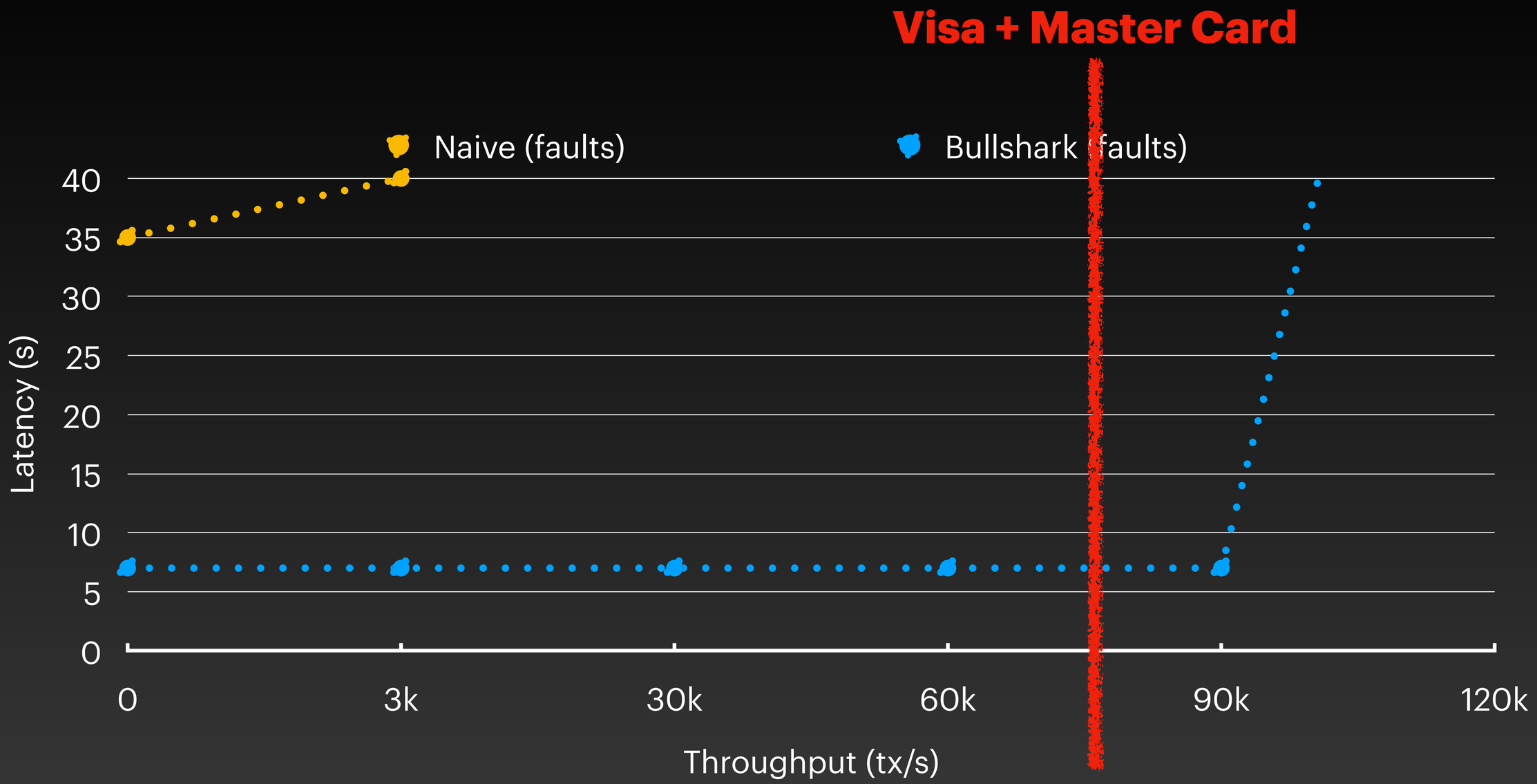

r1
r2
r3
r4
r5
r6
r7
v1
v2
v3
V4

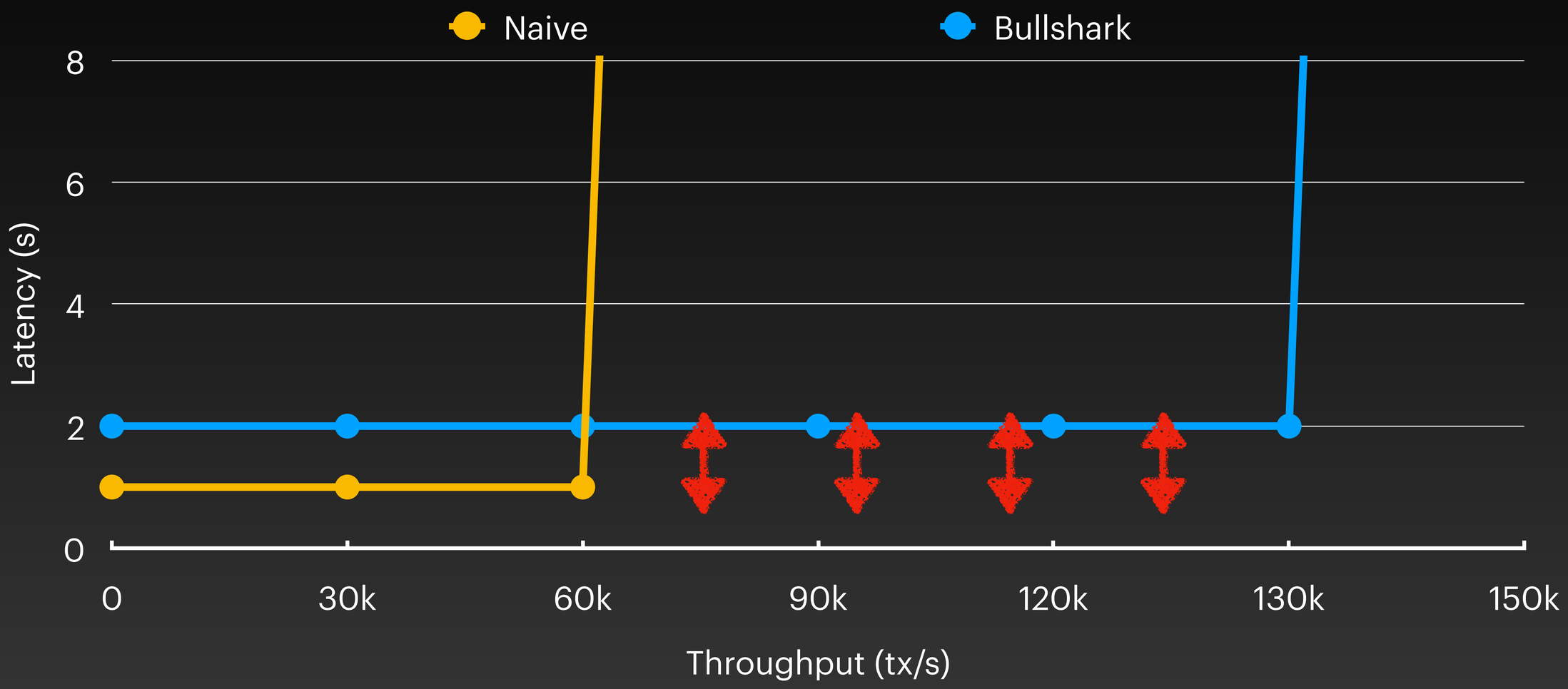
Back Pressure
Robustness

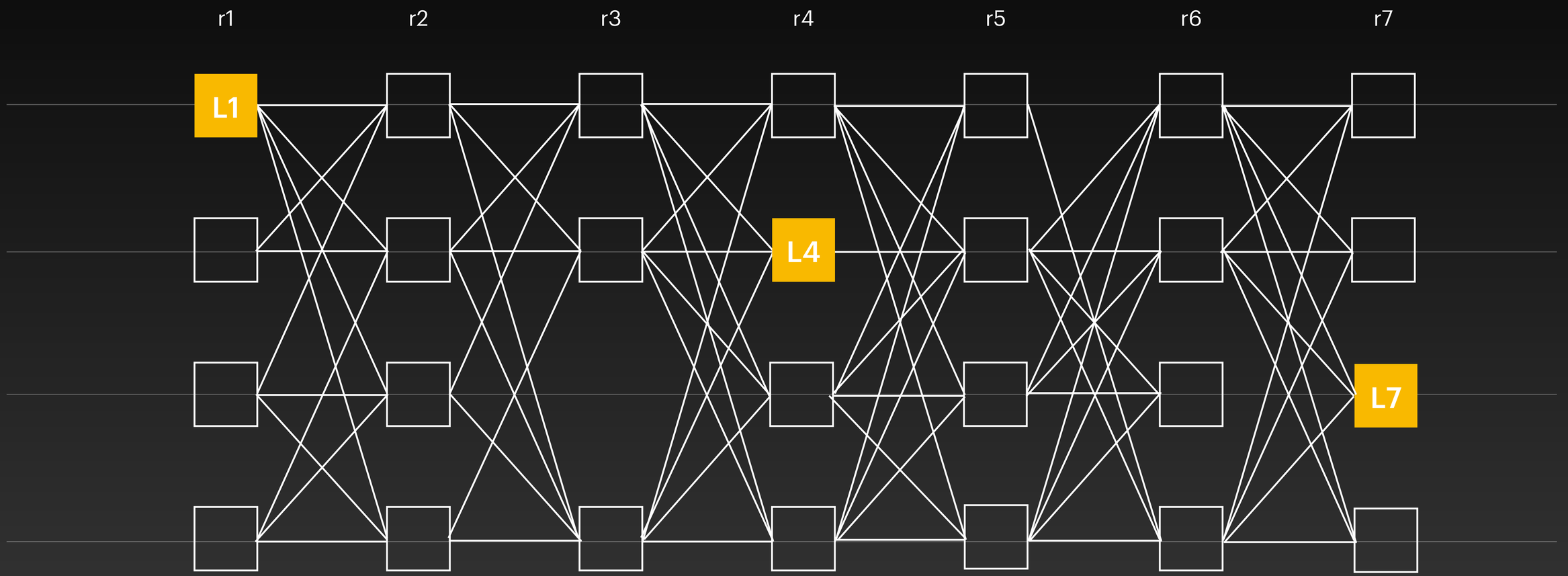
High Resource Utilisation
Throughput


Naive
Bullshark
8
6
4
2
0
Latency (s)
0
30k
60k
90k
120k
130k
150k
Throughput (tx/s)


Naive (faults)
Bullshark (faults)
Latency (s)
40
35
30
25
20
15
10
5
0
0
3k
30k
60k
90k
120k
Throughput (tx/s)


Visa + Master Card
Naive (faults)
Bullshark (faults)
40
35
30
25
20
15
10
5
0
Latency (s)
0
3k
30k
60k
90k
120k
Throughput (tx/s)

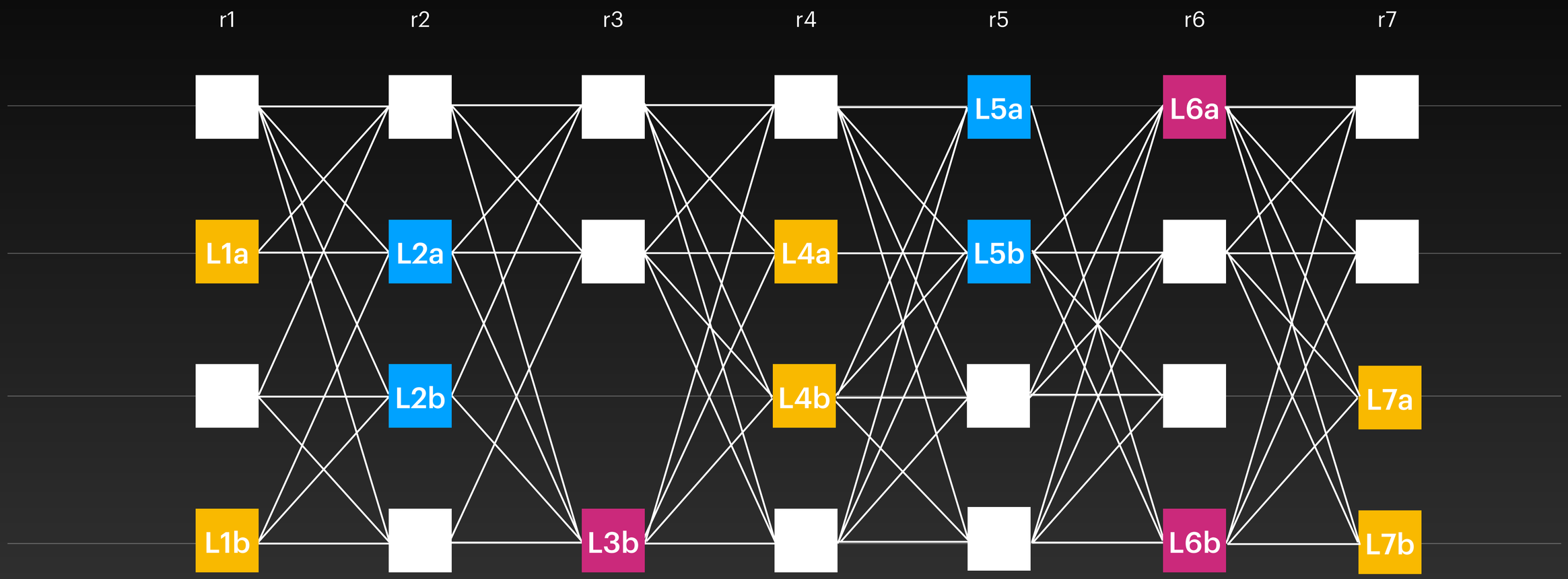

Naive
Bullshark
8
6
4
2
0
Latency (s)
0
30k
60k
90k
120k
130k
150k
Throughput (tx/s)

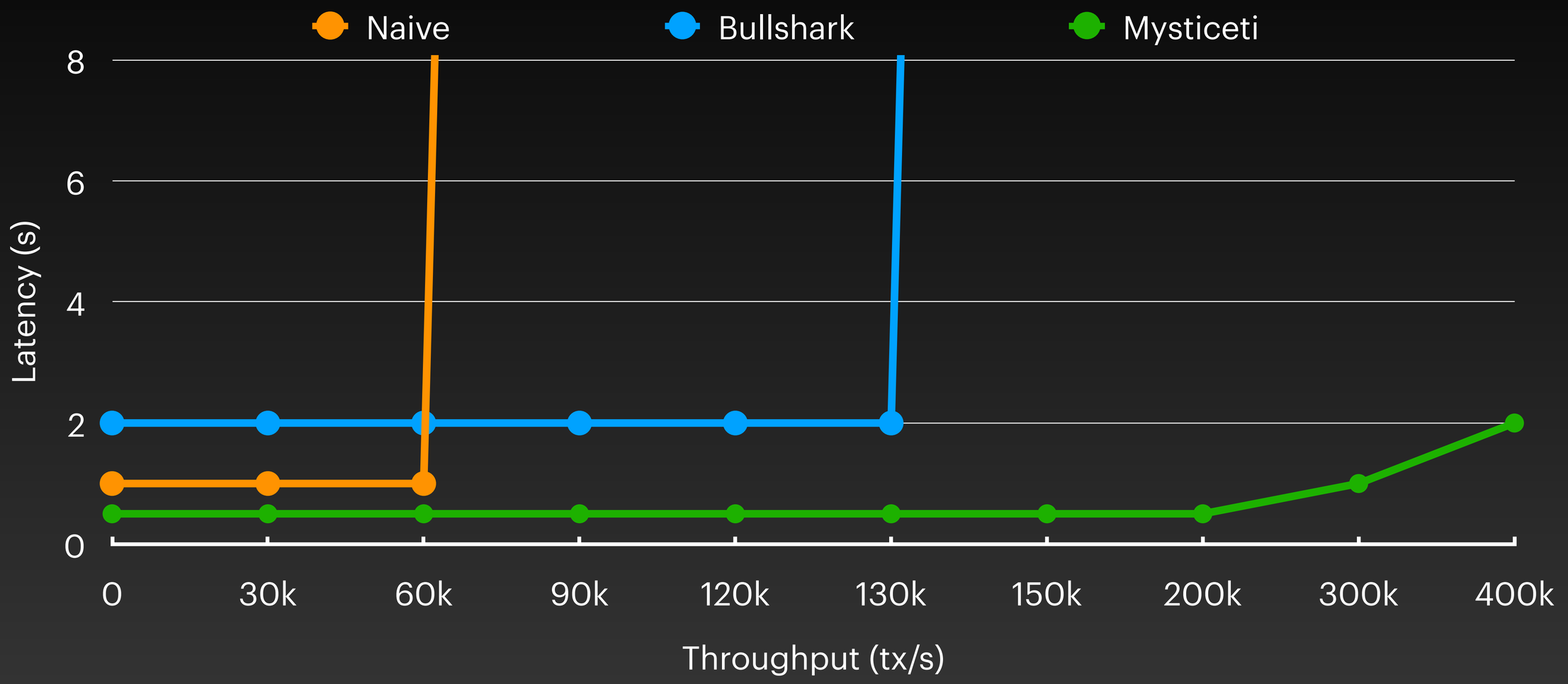

r1
r2
r3
r4
r5
r6
r7
L1
L4
L7

Back Pressure
Robustness

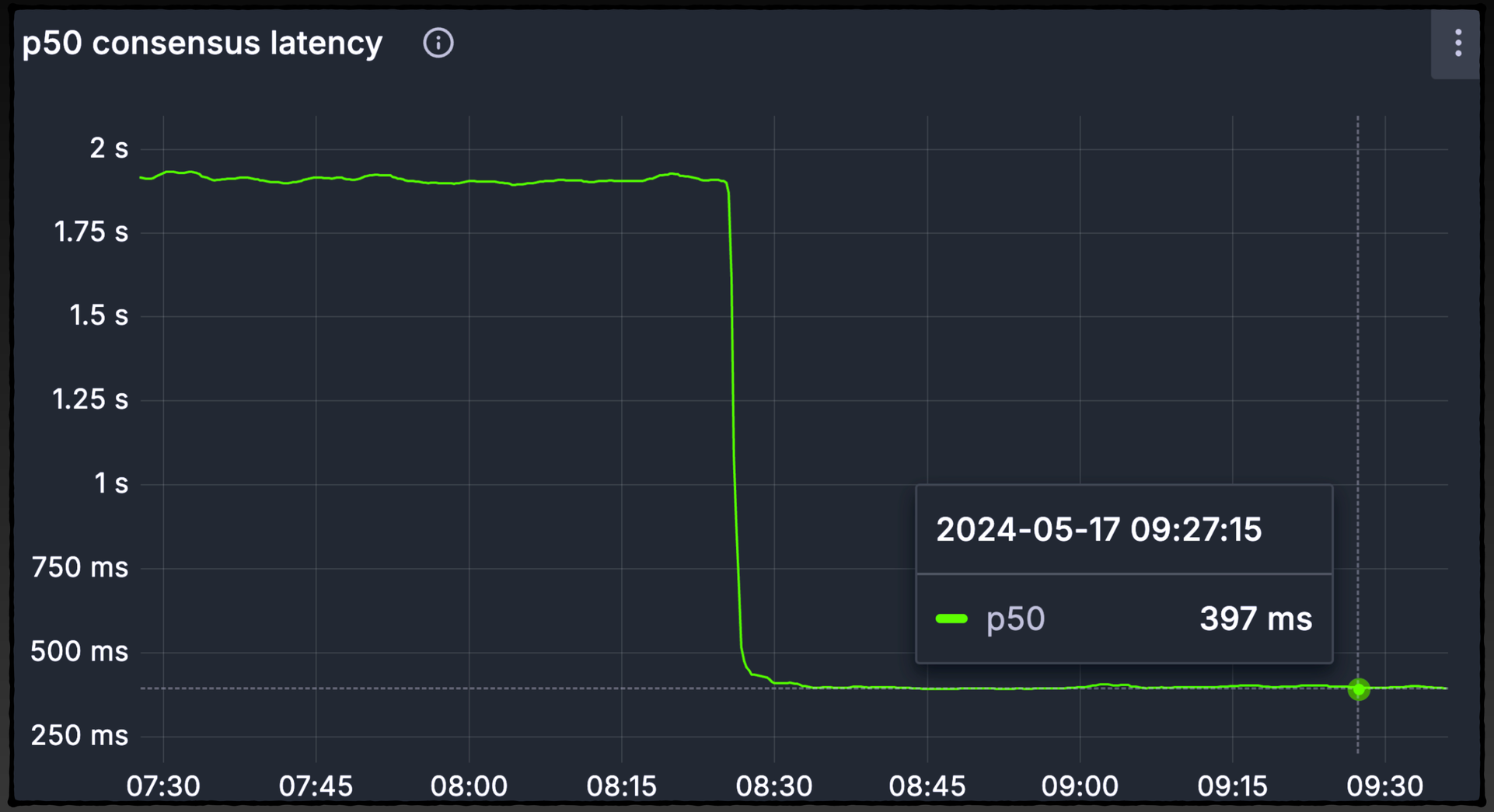
High Resource Utilisation
Throughput

Frequent Commits
Latency


r1
r2
r3
r4
r5
r6
r7
L1a
L1b
L2a
L2b
L3b
L4a
L4b
L5a
L5b
L6a
L6b
L7a
L7b

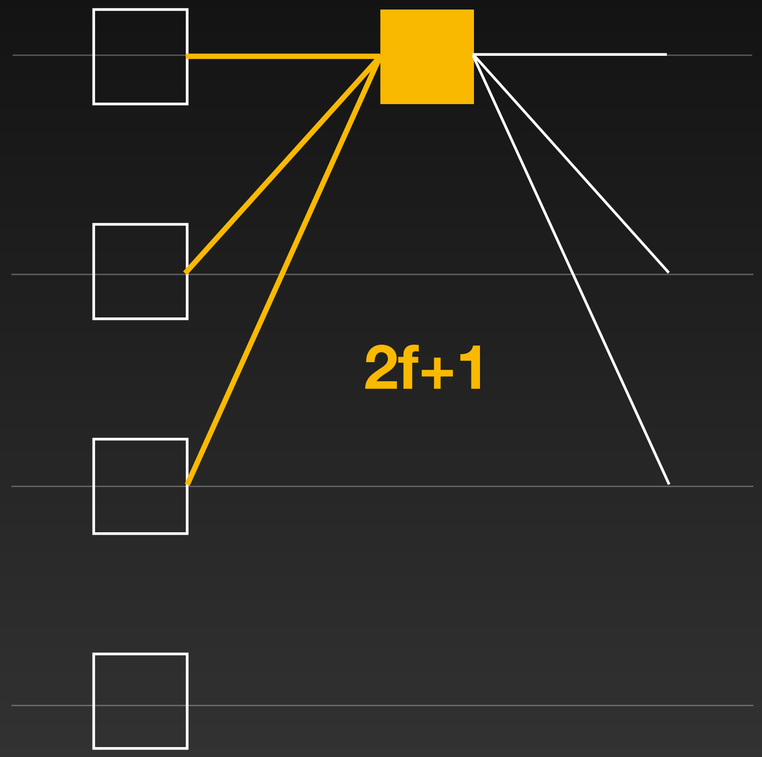

Naive
Bullshark
Mysticeti
Latency (s)
8
6
4
2
0
0
30k
60k
90k
120k
130k
150k
200k
300k
400k
Throughput (tx/s)

The Sui Mainnet

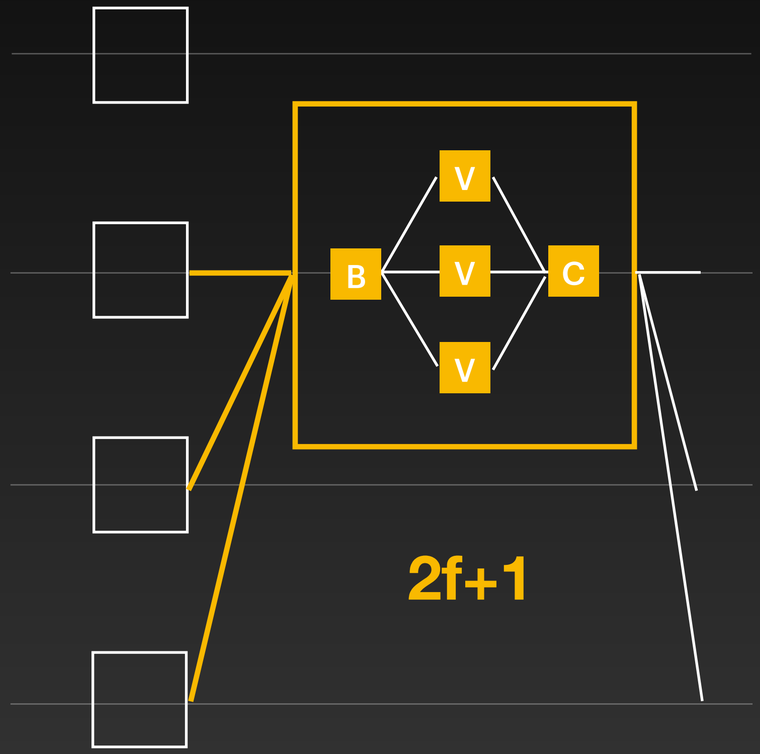


Structured DAGs

Uncertified

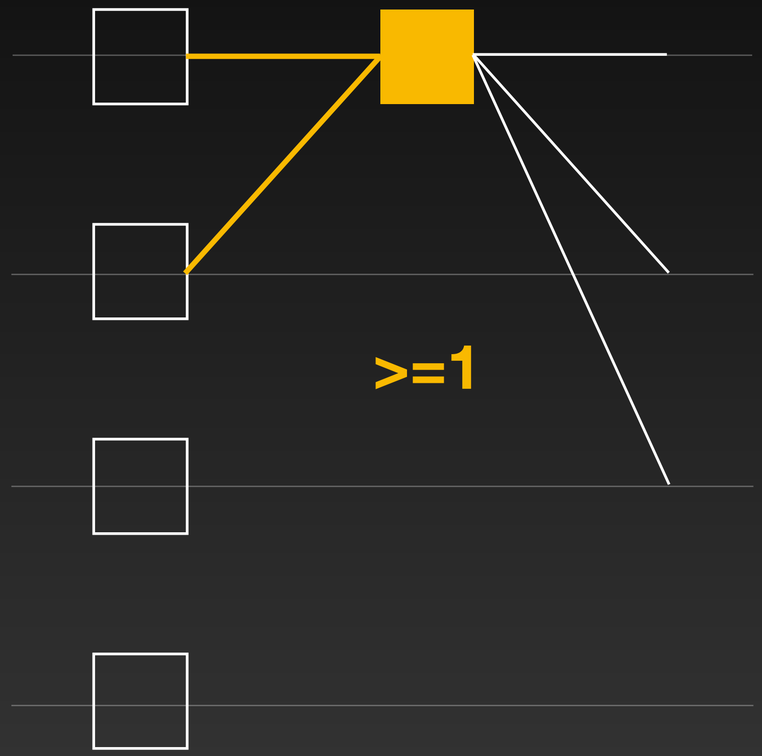


Certified

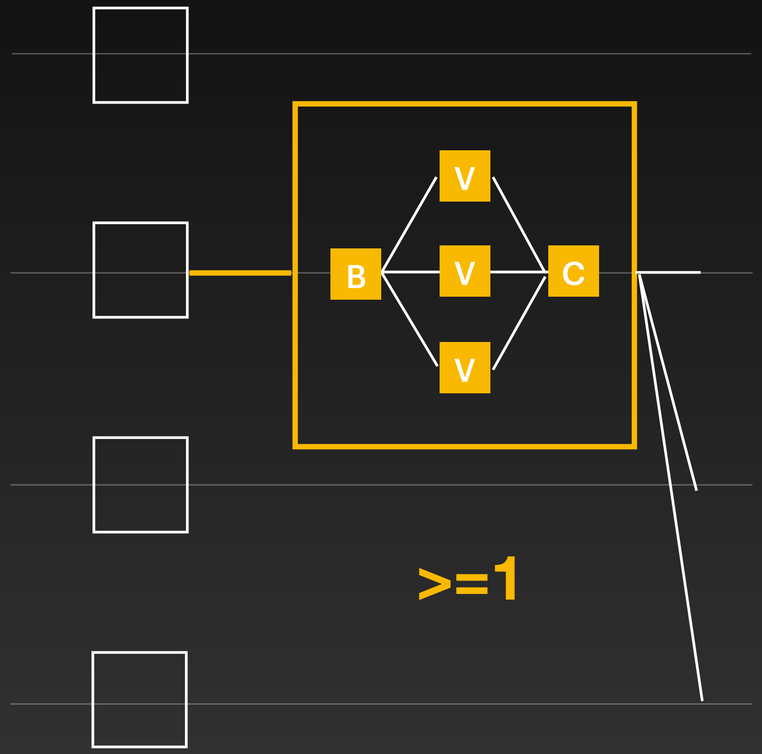


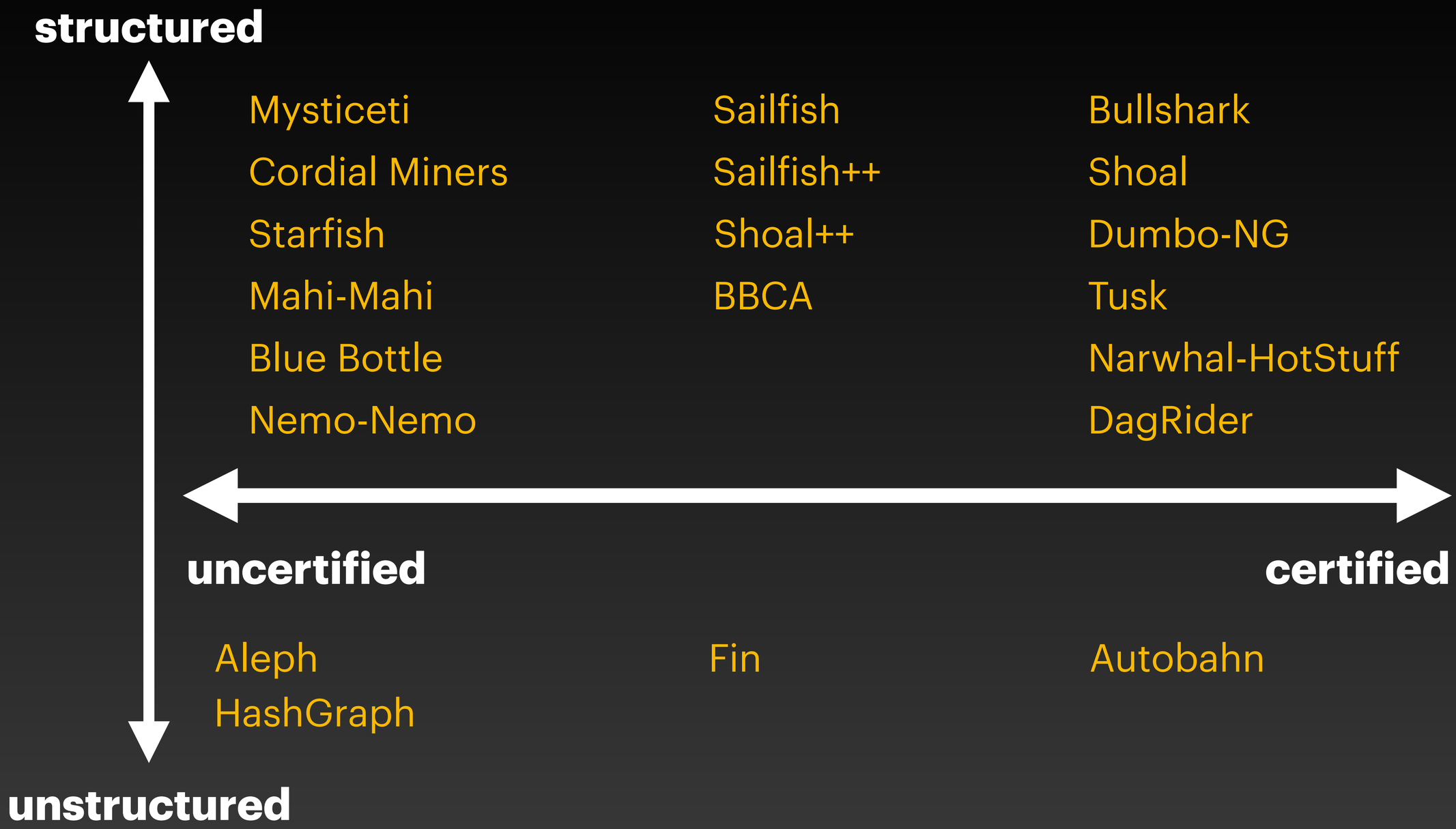
Unstructured DAGs

Uncertified



Certified




structured
Mysticeti
Cordial Miners
Starfish
Mahi-Mahi
Blue Bottle
Nemo-Nemo
Sailfish
Sailfish++
Shoal++
BBCA
Bullshark
Shoal
Dumbo-NG
Tusk
Narwhal-HotStuff
DagRider
uncertified
certified
Aleph
HashGraph
Fin
Autobahn
unstructured

Certified
High Robustness • Easier Block Synchronisation

Uncertified
Low CPU • Graceful Crash Faults

Unstructured
Network Flexibility

Cordial Miners

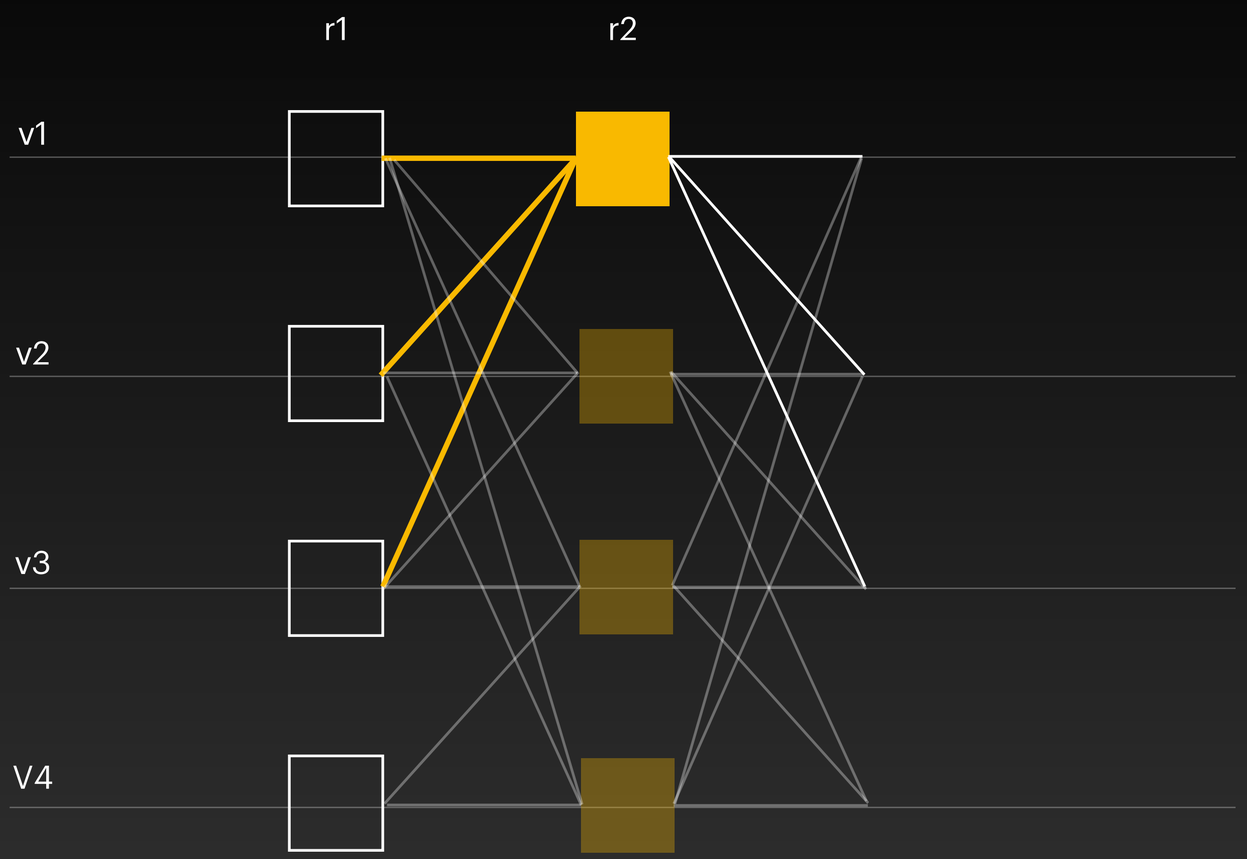
Bullshark

Shoal++

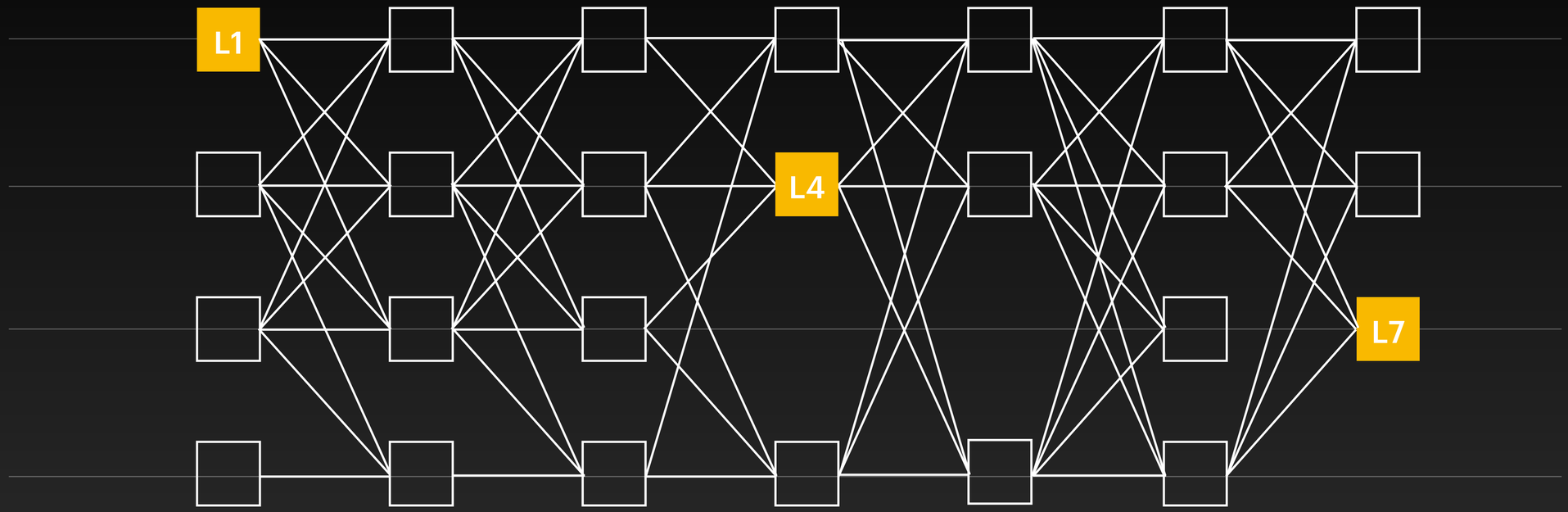
Mysticeti

…

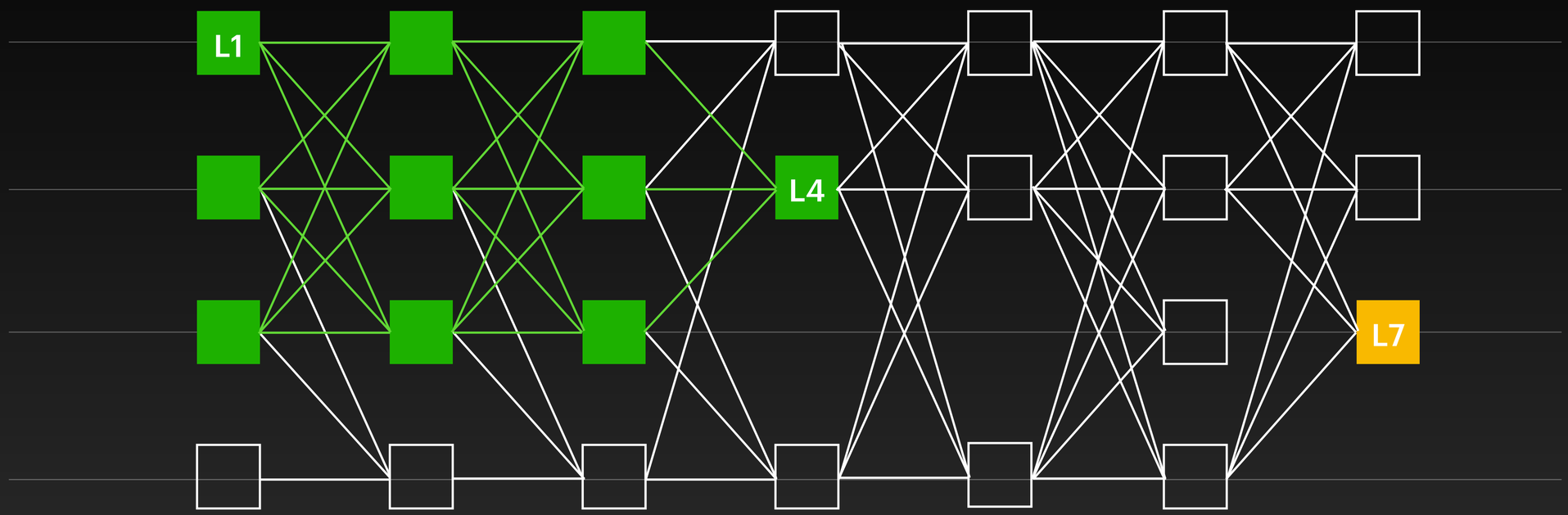
Structured • Uncertified



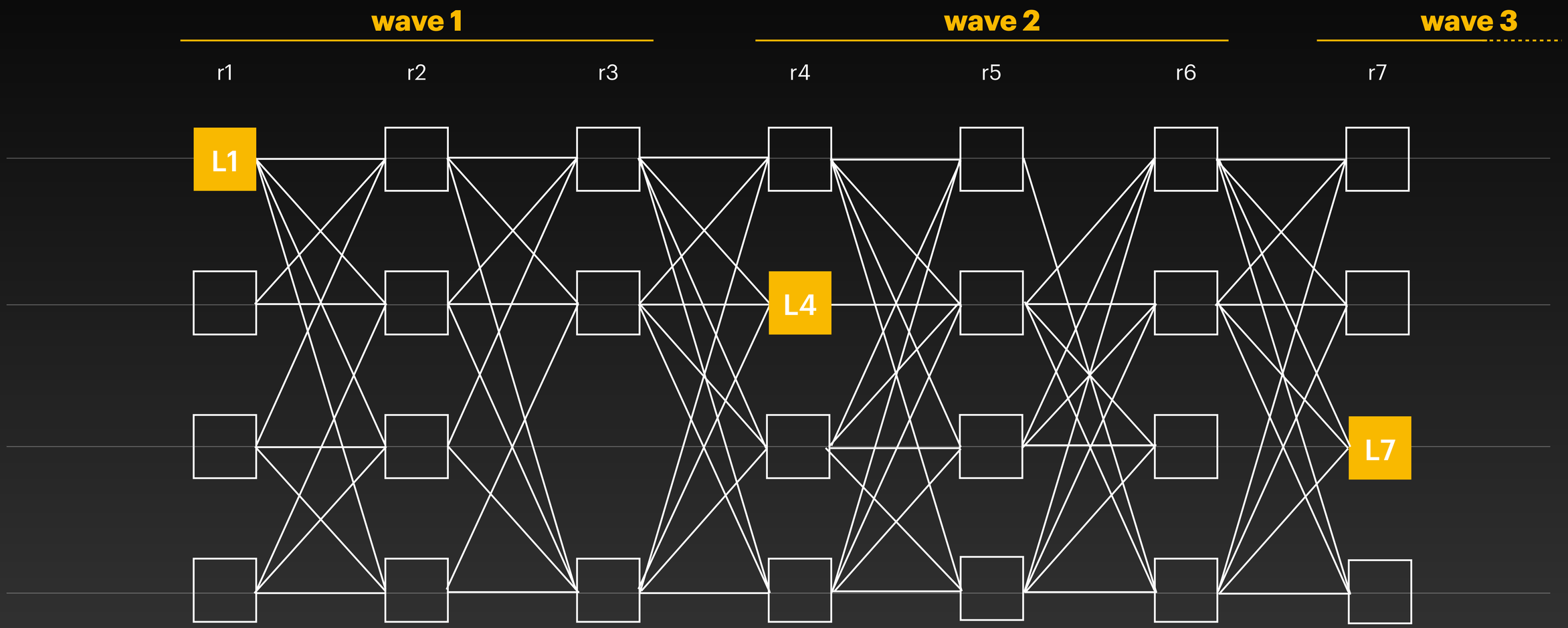
- Round number
- Author
- Payload (txs)
- **2f+1 Parents**
- Signature

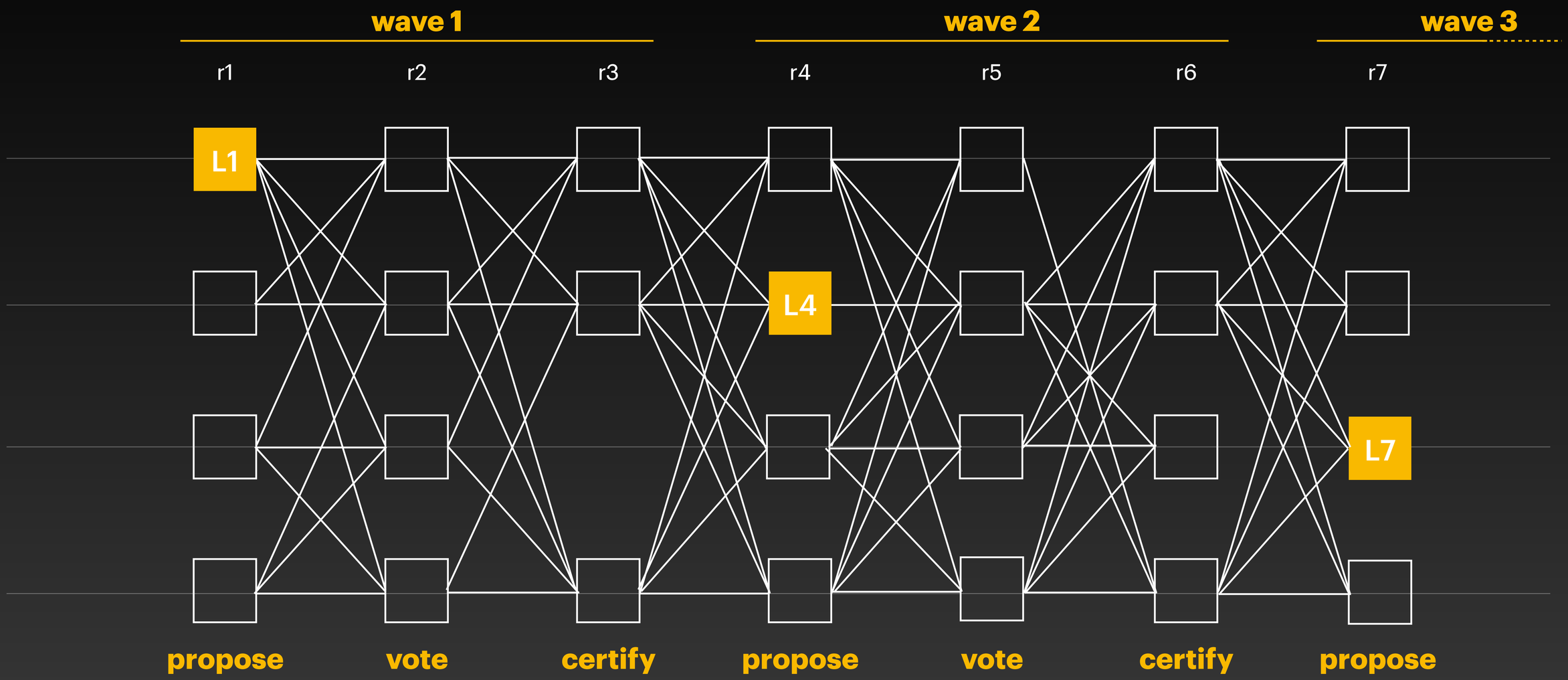


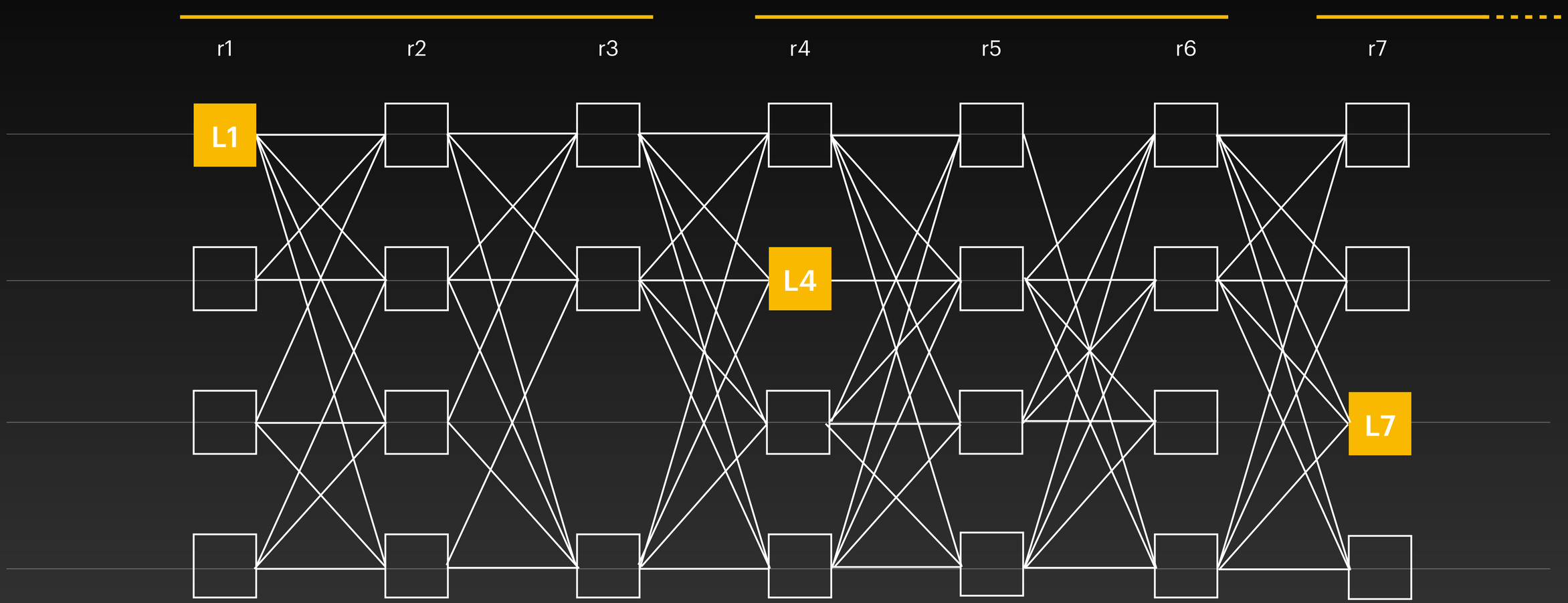
- We focus on ordering leaders: L1 L4 L7

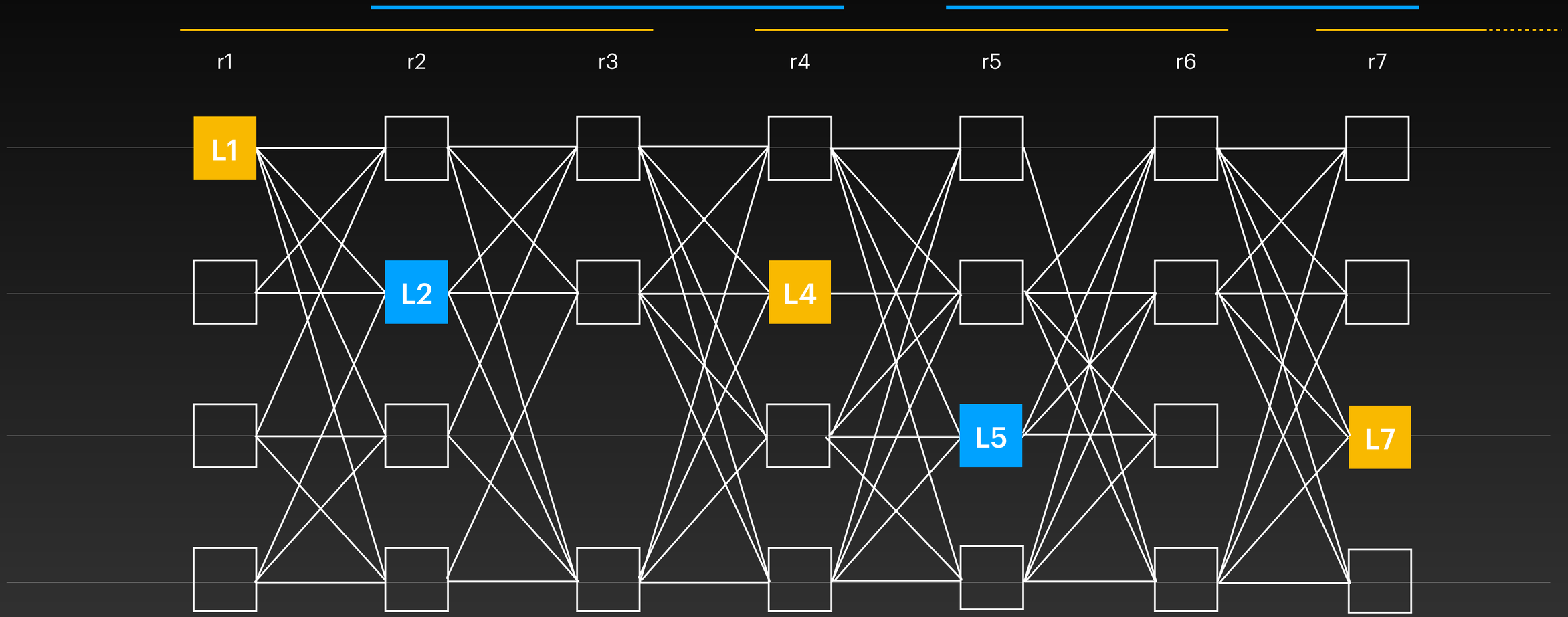


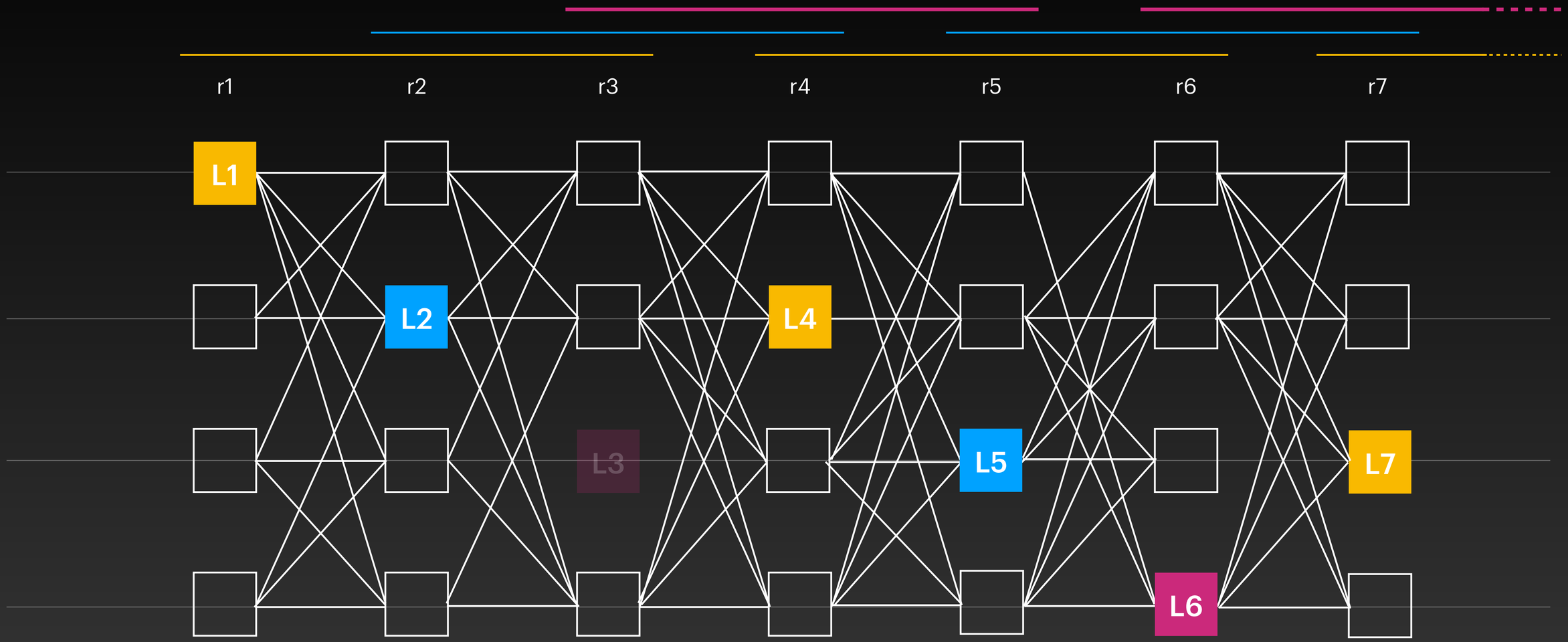
- We focus on ordering leaders: L1 L4 L7
- Linearising the sub-DAG is simple

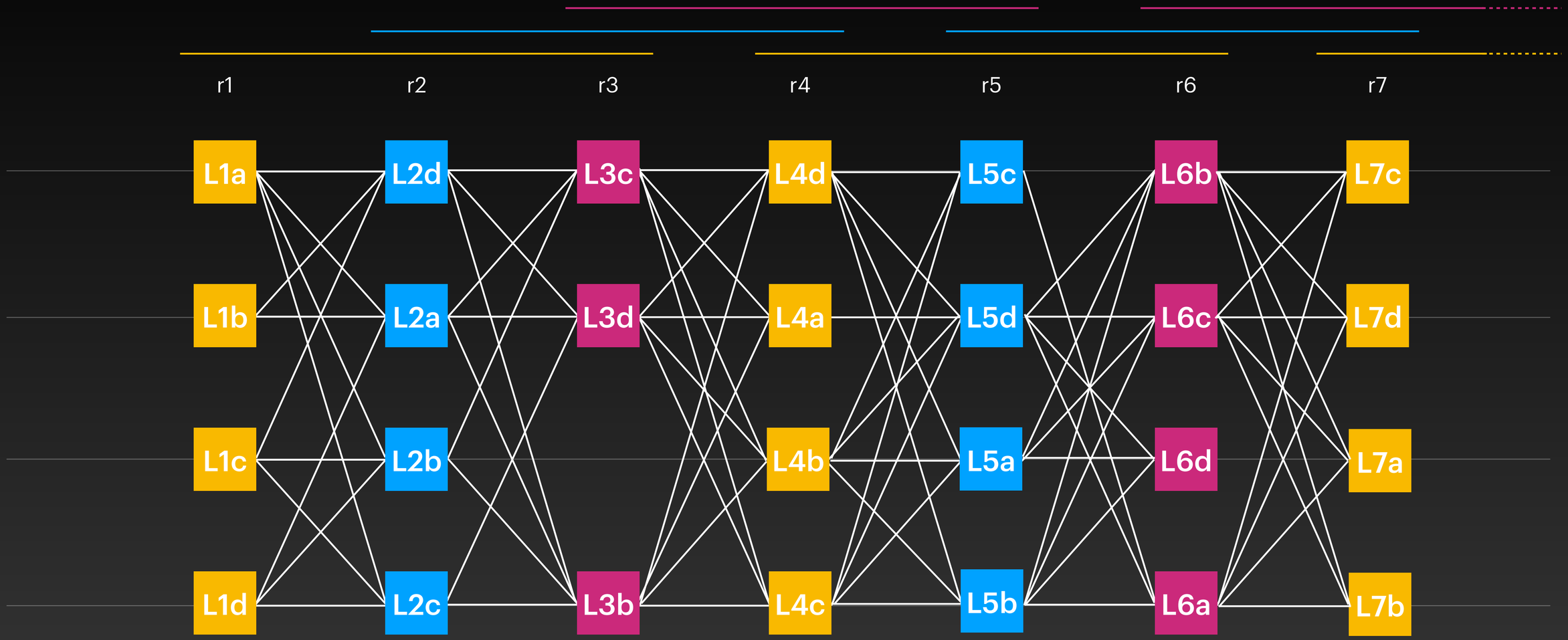

wave 1
wave 2
wave 3
r1
r2
r3
r4
r5
r6
r7
L1
L4
L7

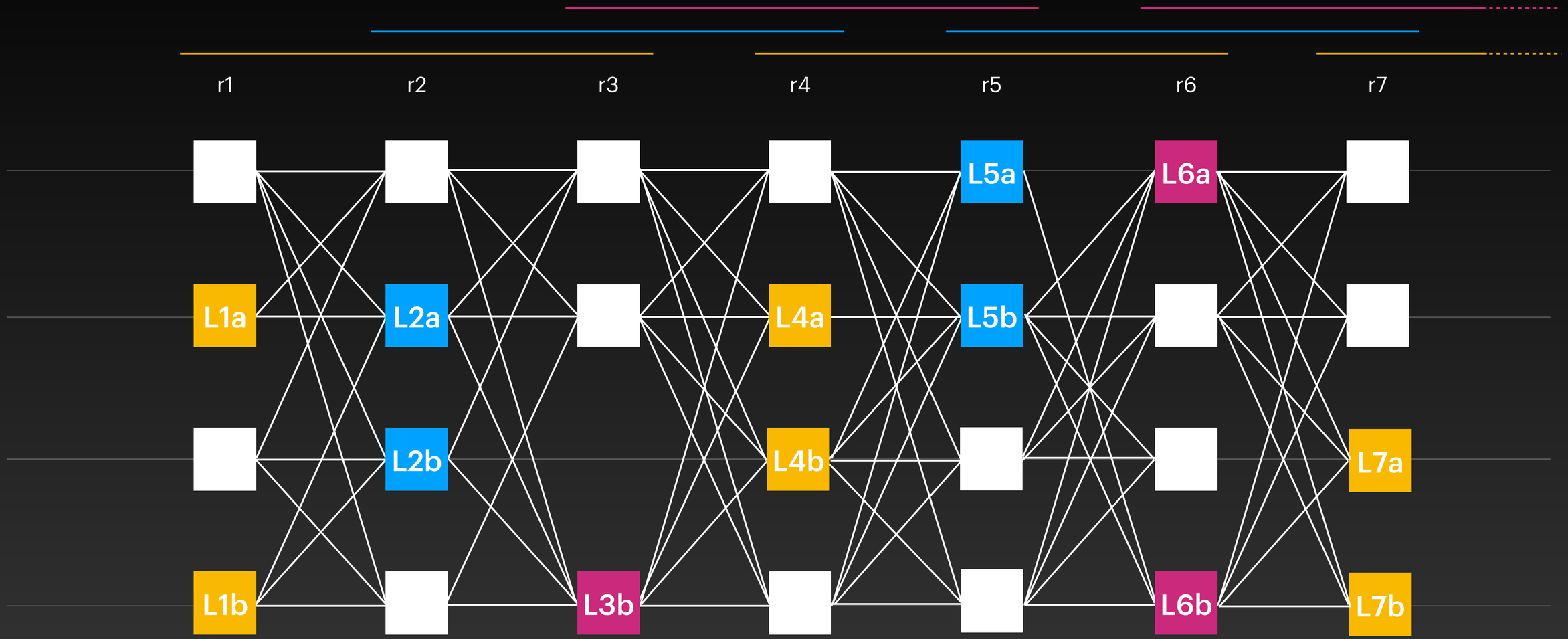

wave 1
wave 2
wave 3
r1
r2
r3
r4
r5
r6
r7
L1
L4
L7
propose
vote
certify
propose
vote
certify
propose

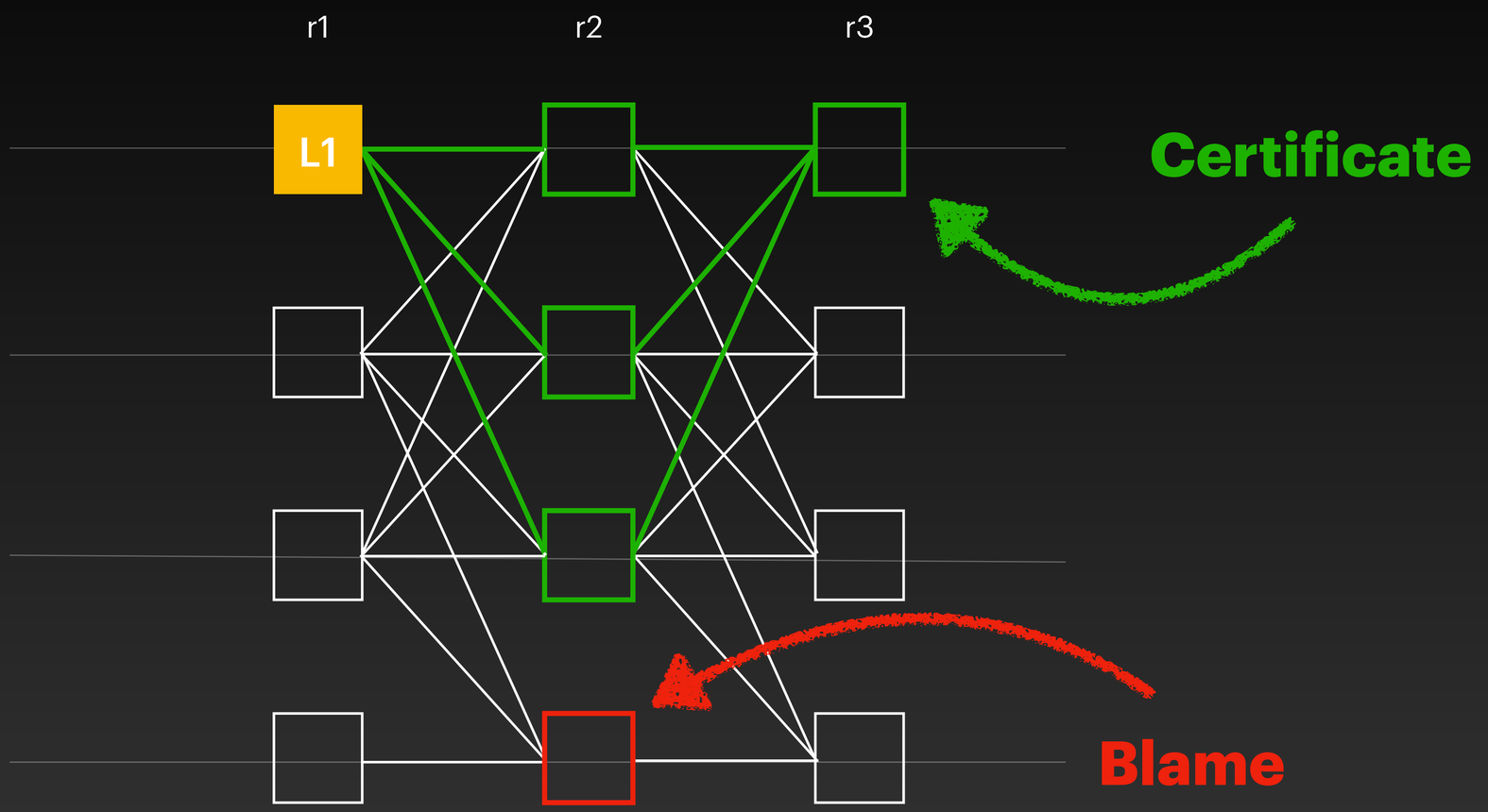

r1
r2
r3
r4
r5
r6
r7
L1
L4
L7


r1
r2
r3
r4
r5
r6
r7
L1
L2
L4
L5
L7


r1
r2
r3
r4
r5
r6
r7
L1
L2
L3
L4
L5
L6
L7


r1
r2
r3
r4
r5
r6
r7
L1a
L1b
L1c
L1d
L2d
L2a
L2b
L2c
L3c
L3d
L3b
L4d
L4a
L4b
L4c
L5c
L5d
L5a
L5b
L6b
L6c
L6d
L6a
L7c
L7d
L7a
L7b


r1
r2
r3
r4
r5
r6
r7
L1a
L1b
L2a
L2b
L3b
L4a
L4b
L5a
L5b
L6a
L6b
L7a
L7b


r1
r2
r3
L1
Certificate
Blame

Direct Decision Rule

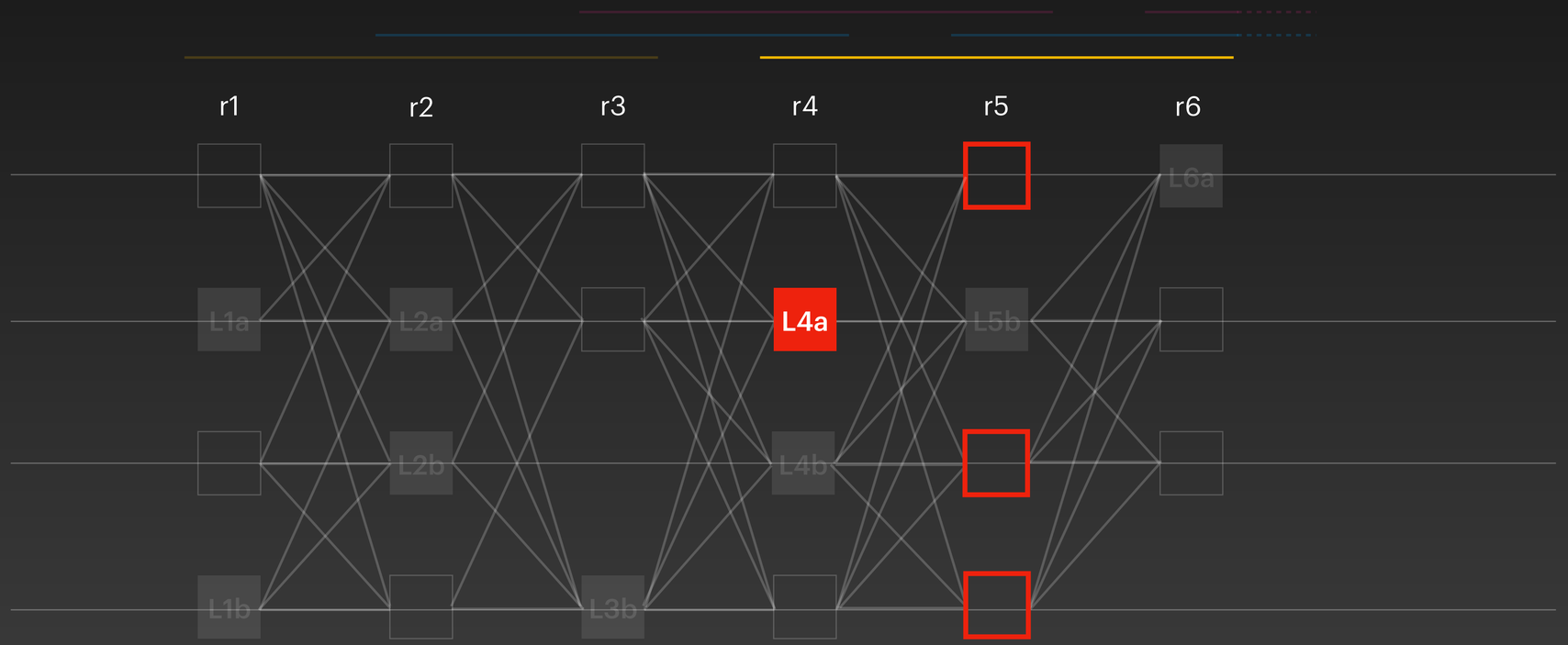
On each leader starting from highest round:

- **Skip** if 2f+1 blames
- **Commit** if 2f+1 certificates
- **Undecided** otherwise

Direct Decision Rule

On each leader starting from highest round:

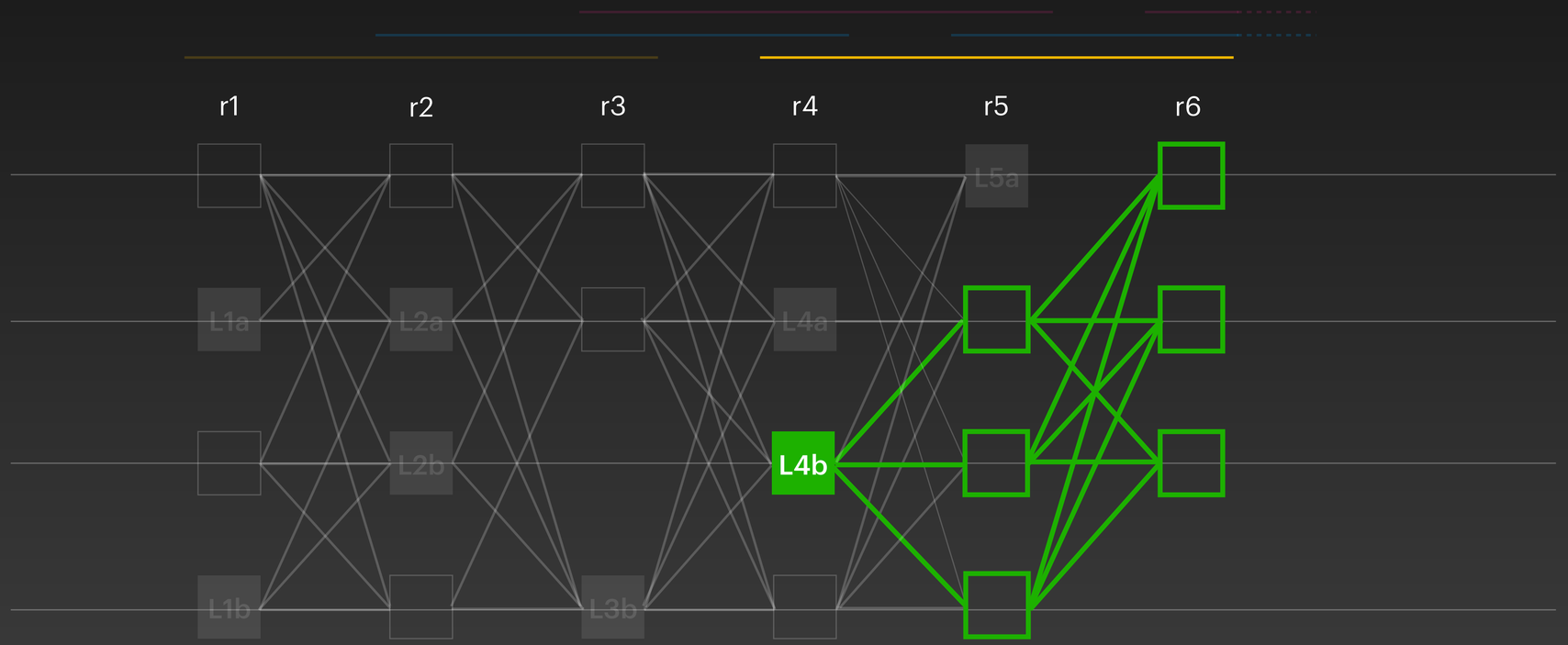
- **Skip** if 2f+1 blames
- **Commit** if 2f+1 certificates
- **Undecided** otherwise



Direct Decision Rule

On each leader starting from highest round:

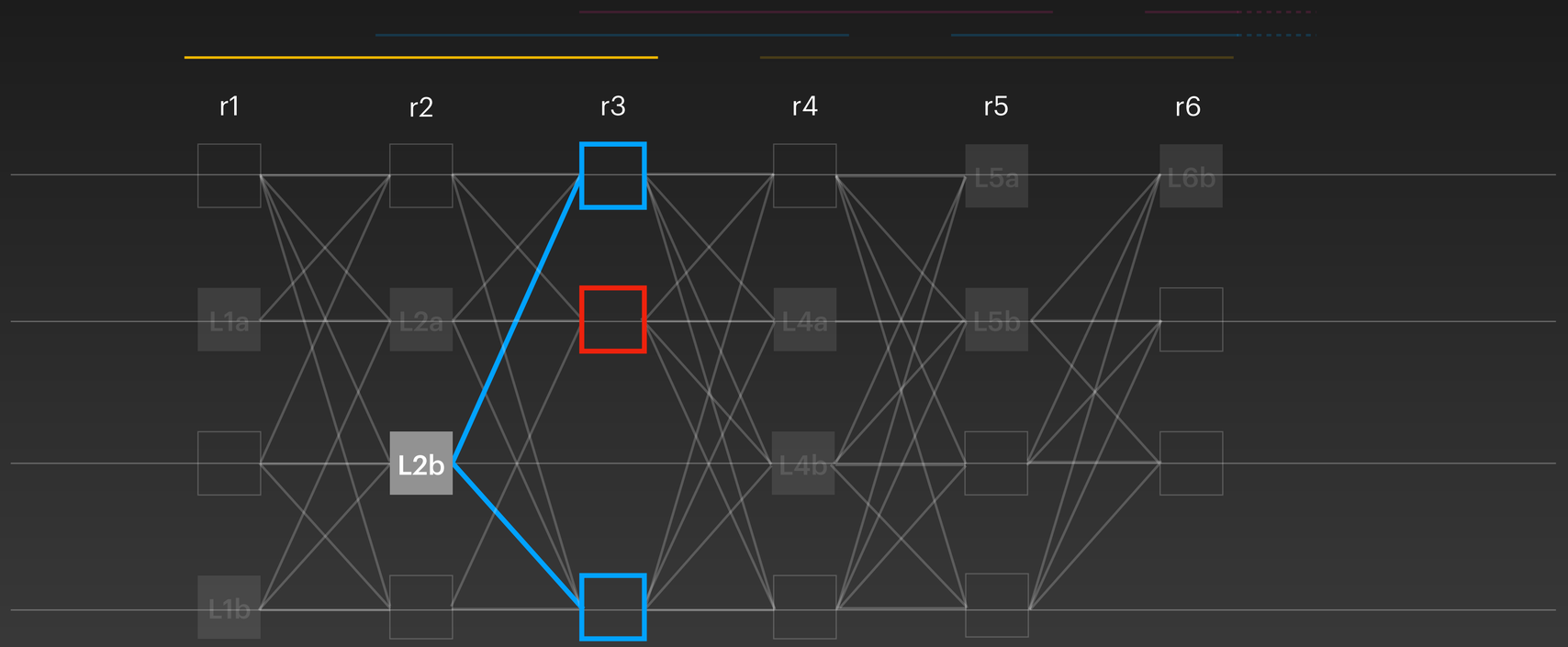
- **Skip** if 2f+1 blames
- **Commit** if 2f+1 certificates
- **Undecided** otherwise



Direct Decision Rule

On each leader starting from highest round:

- **Skip** if 2f+1 blames
- **Commit** if 2f+1 certificates
- **Undecided** otherwise

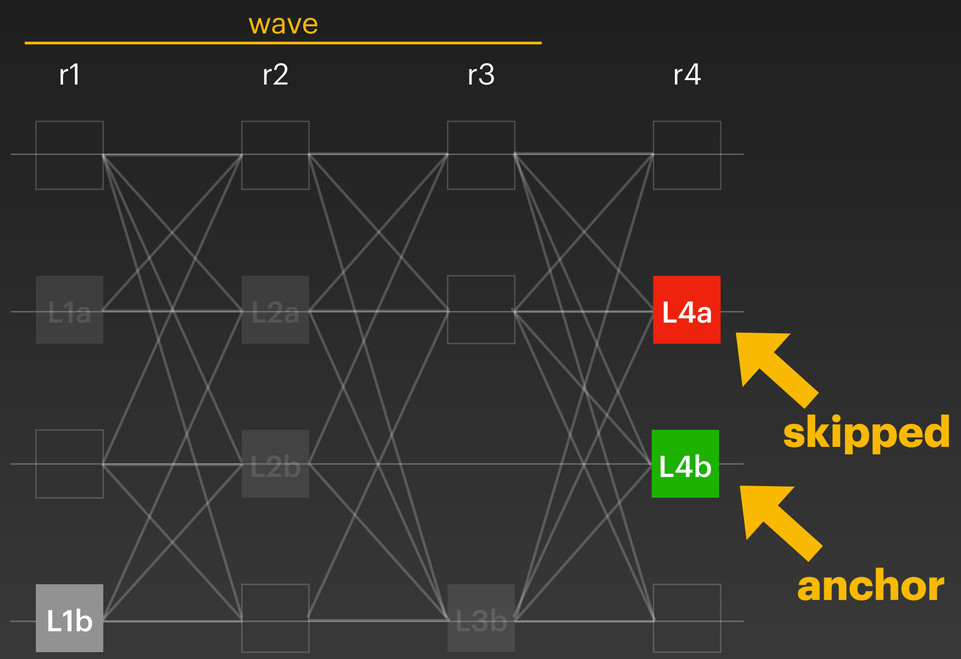


Indirect Decision Rule

Indirect Decision Rule

1. Find Anchor

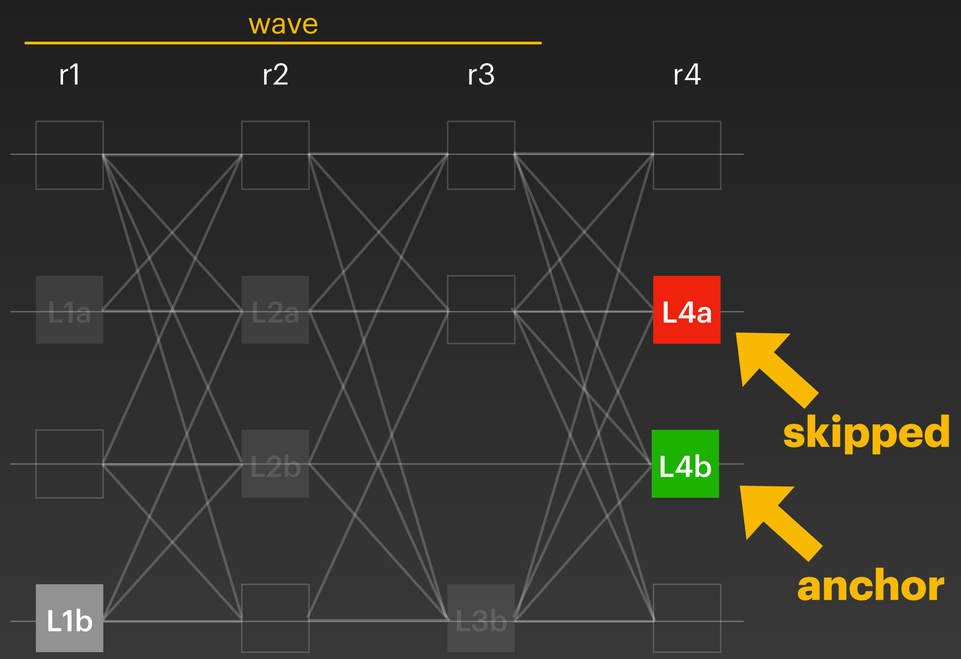
- First block with round > r+2 that is **Commit** or **Undecided**



Indirect Decision Rule

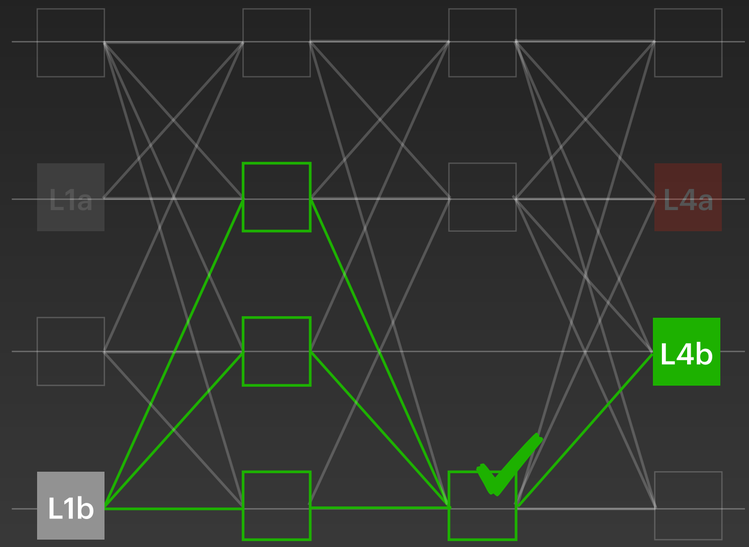
1. Find Anchor

- First block with round > r+2 that is **Commit** or **Undecided**

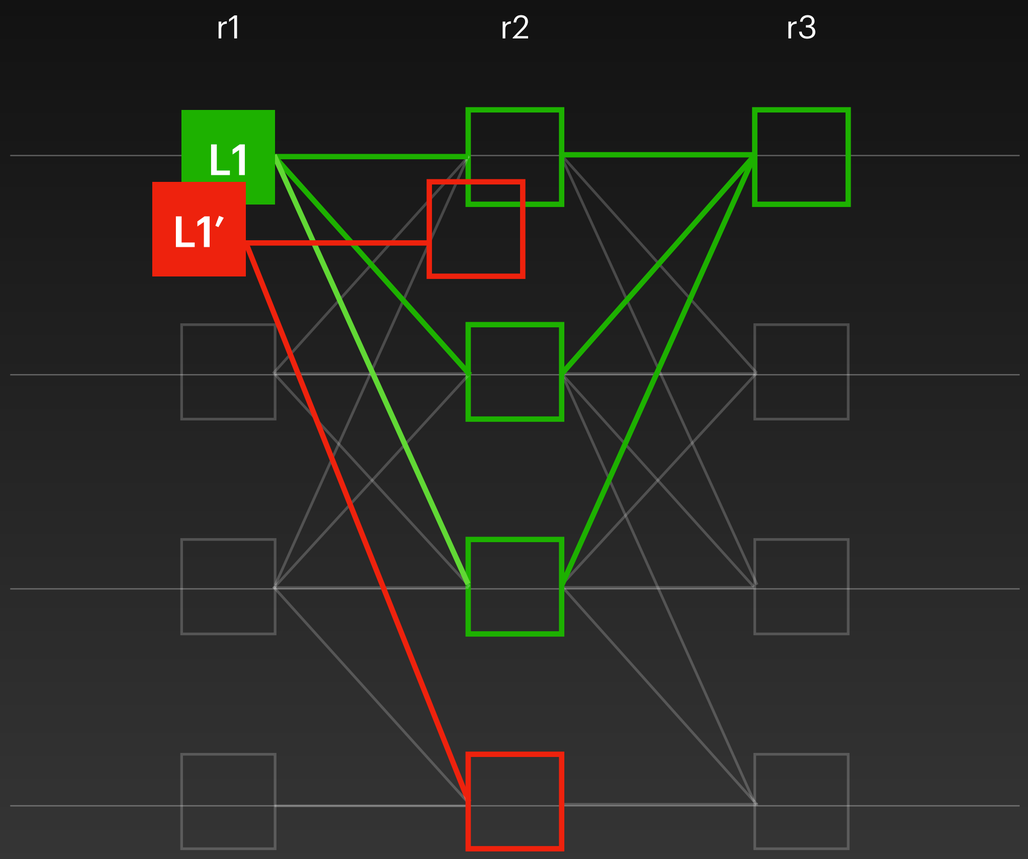


2. Certified link

- **Commit** if `B <-> certified link <-> A` otherwise **Skip**

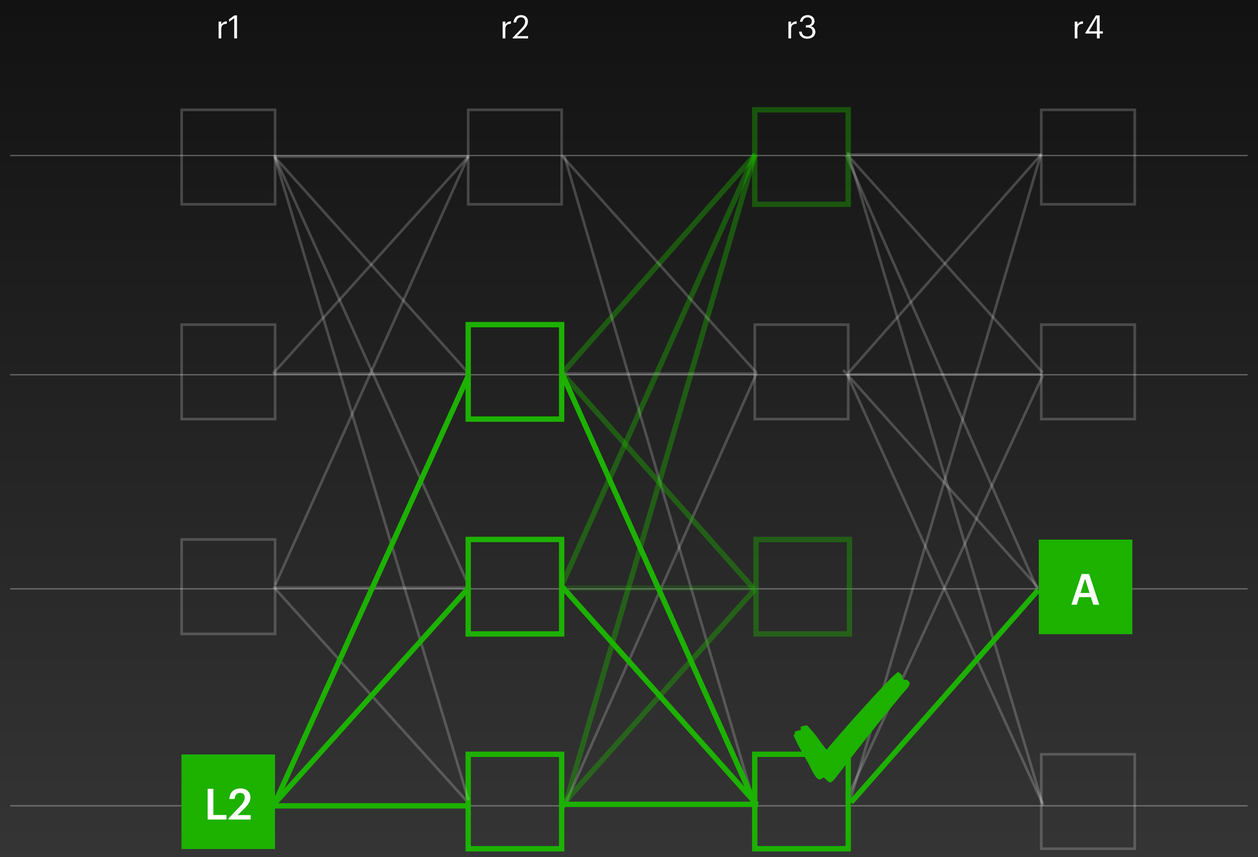


Safety



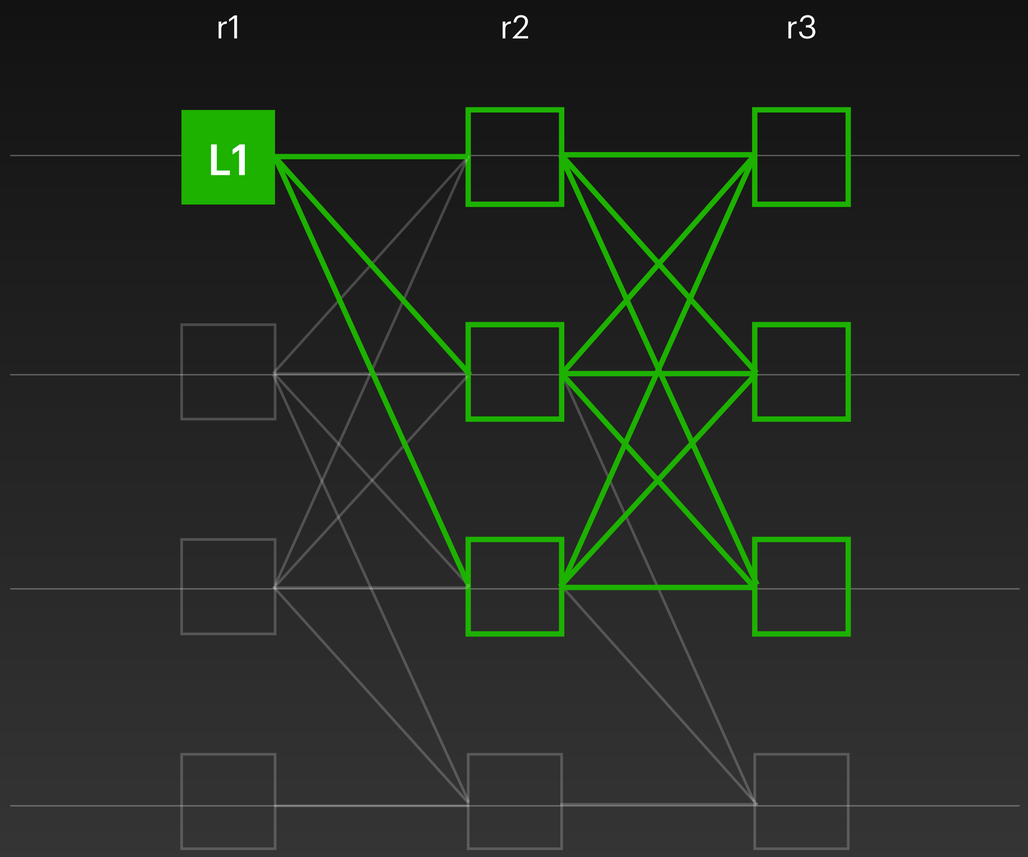
- At most **L1** or **L1’** can have a certificate pattern (quorum intersection)

Safety



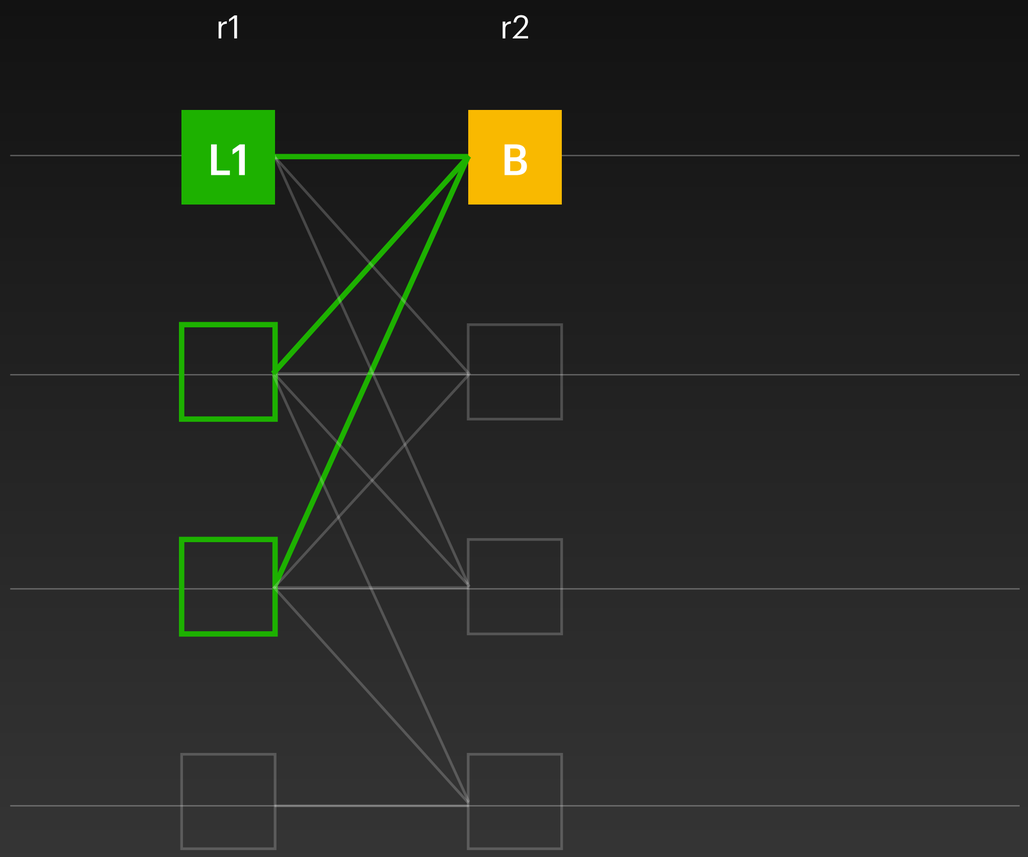
- At most **L1** or **L1'** can have a certificate pattern (quorum intersection
- If **L2** has 2f+1 certificate patterns, **A** always has a certified link to **L2**

Liveness



- After GST, the direct decision rule **commits** a block

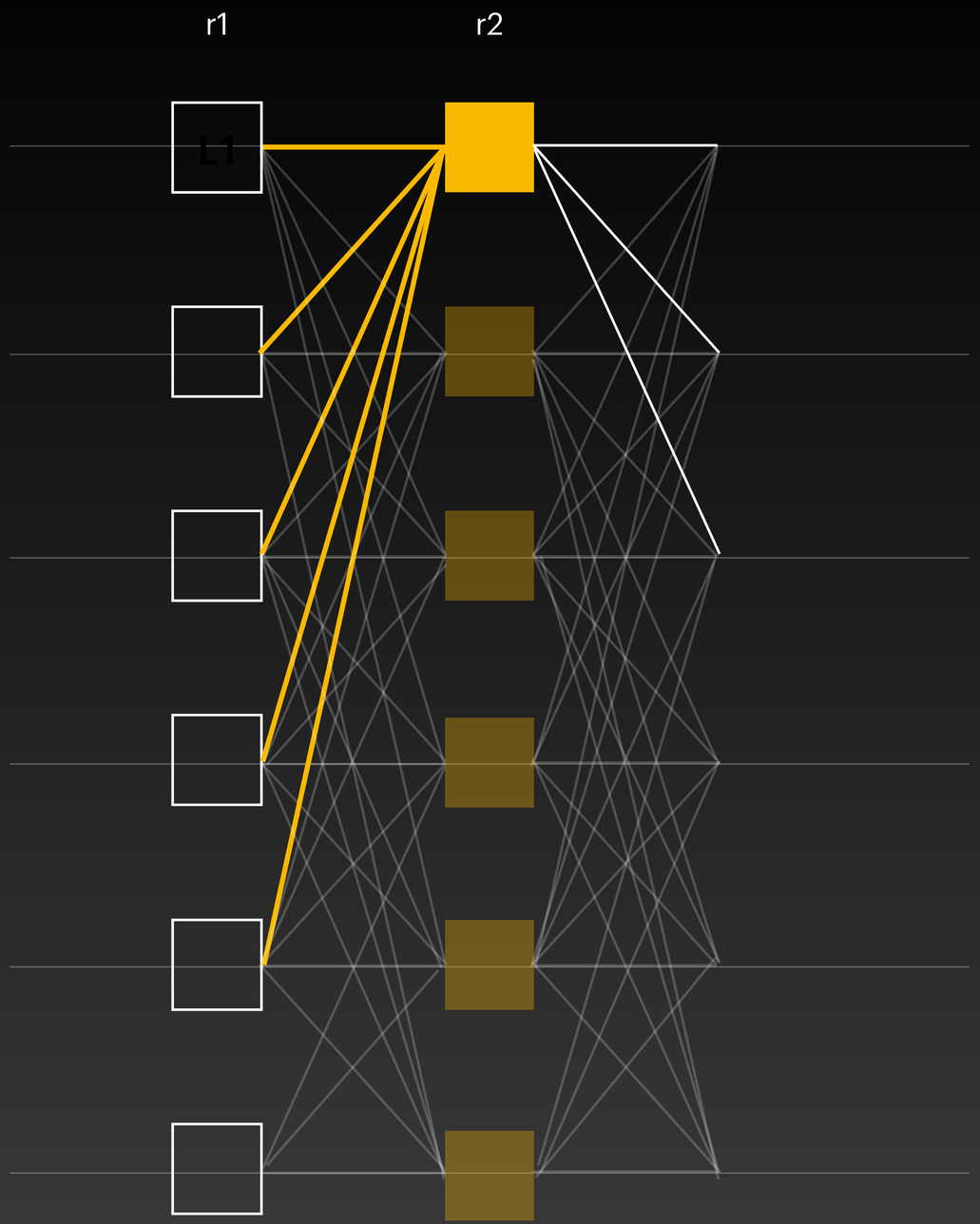
Liveness



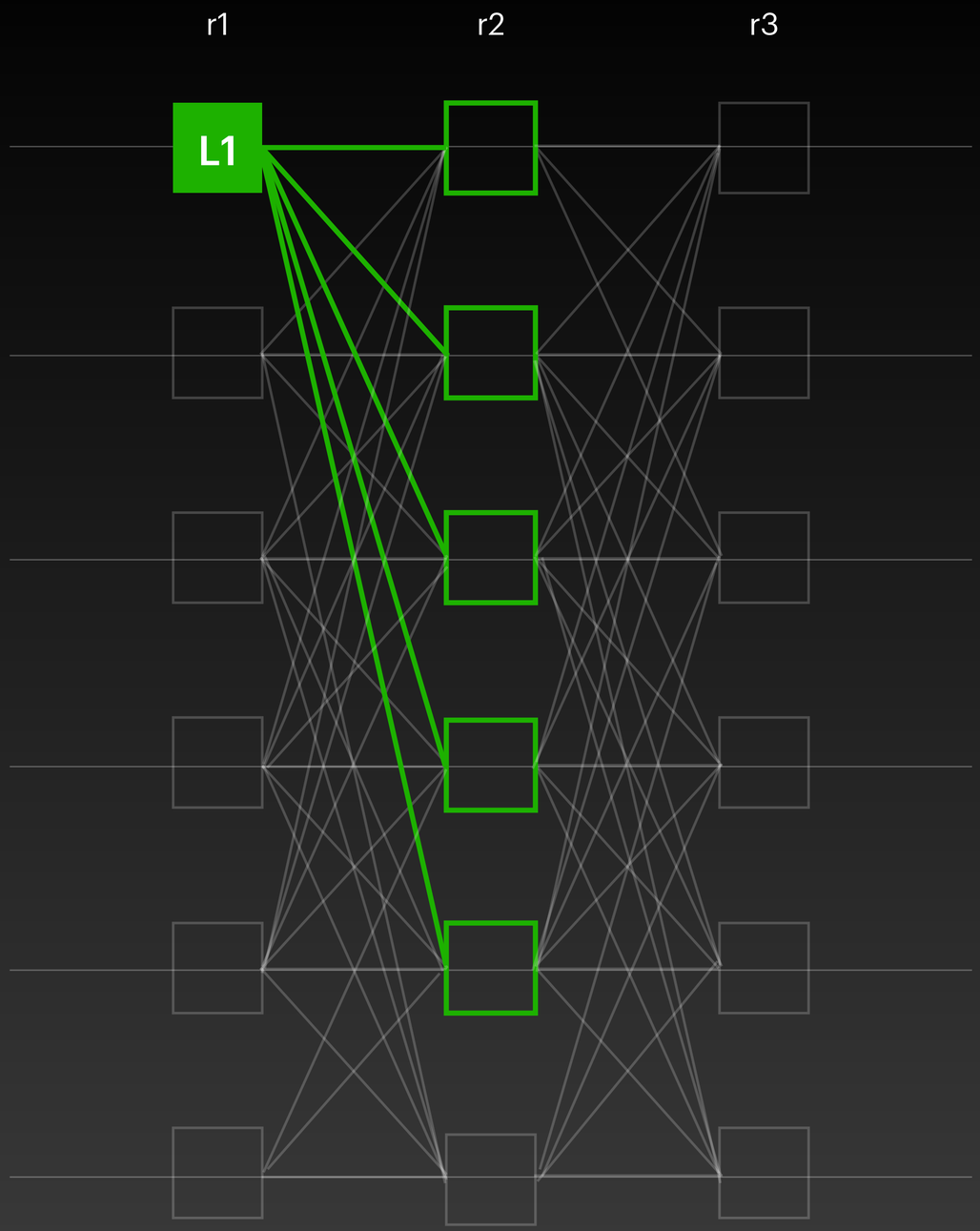
- After GST, the direct decision rule **commits** a block
- Leader timeout: **B** waits for 2f+1 parents + 250 ms

- 5f+1 vs 3f+1 (higer quaroumm vs less round; queuing latency remains the same
- What devices is geo-location; 5f+1 is too far and liveness goes down
- 5f+3c+1 + fault_coverage.pdf
- Graphs 10k, 100k, percentage improvement

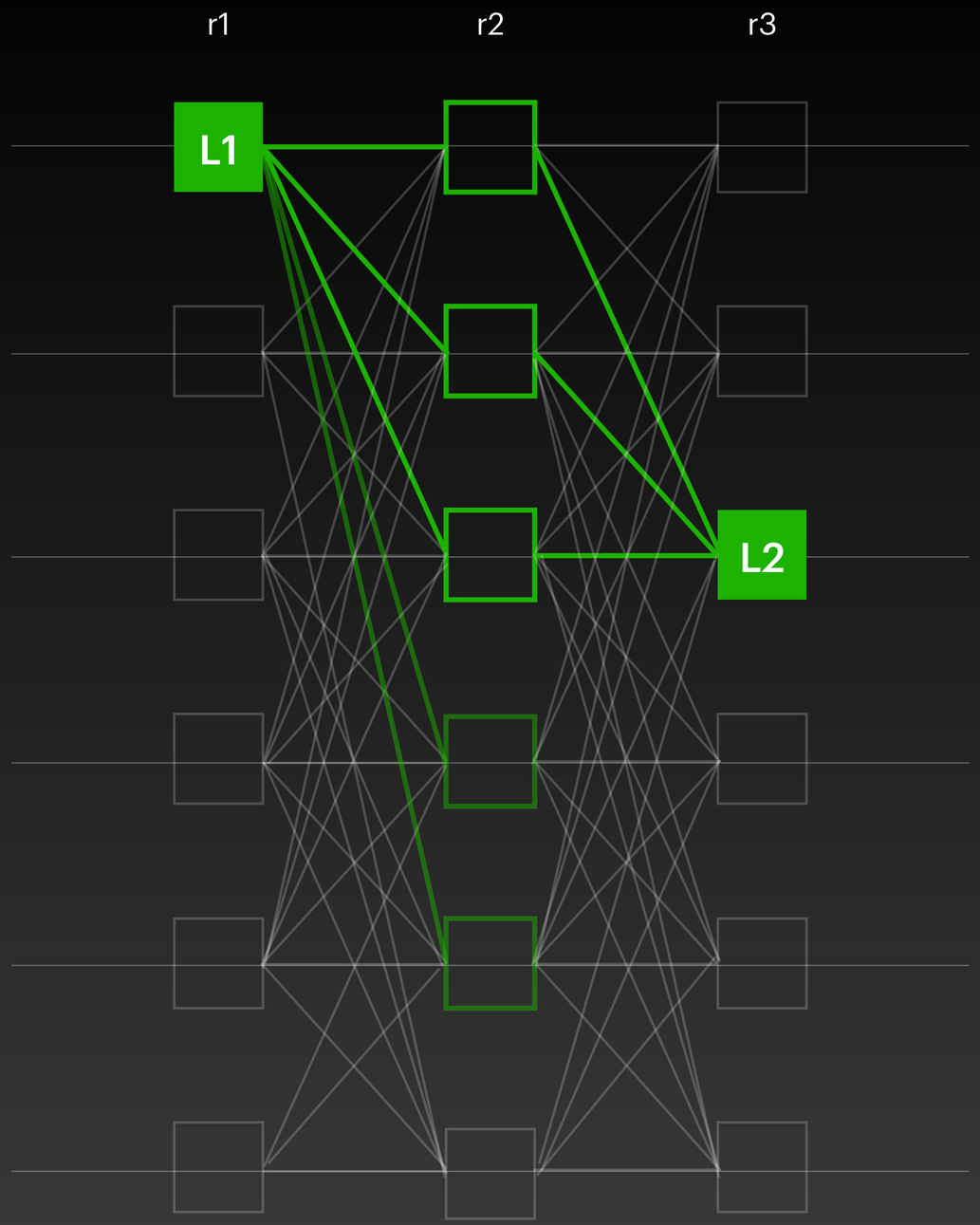
5f+1



- Round number
- Author
- Payload (txs)
- **4f+1 Parents**
- Signature



- **Direct commit:** At least **4f+1** votes for **L1**



- **Direct commit:** At least **4f+1** votes for **L1**
- **Indirect commit:** At least **2f+1** links from **L2** to **L1**

Quorum Sizes

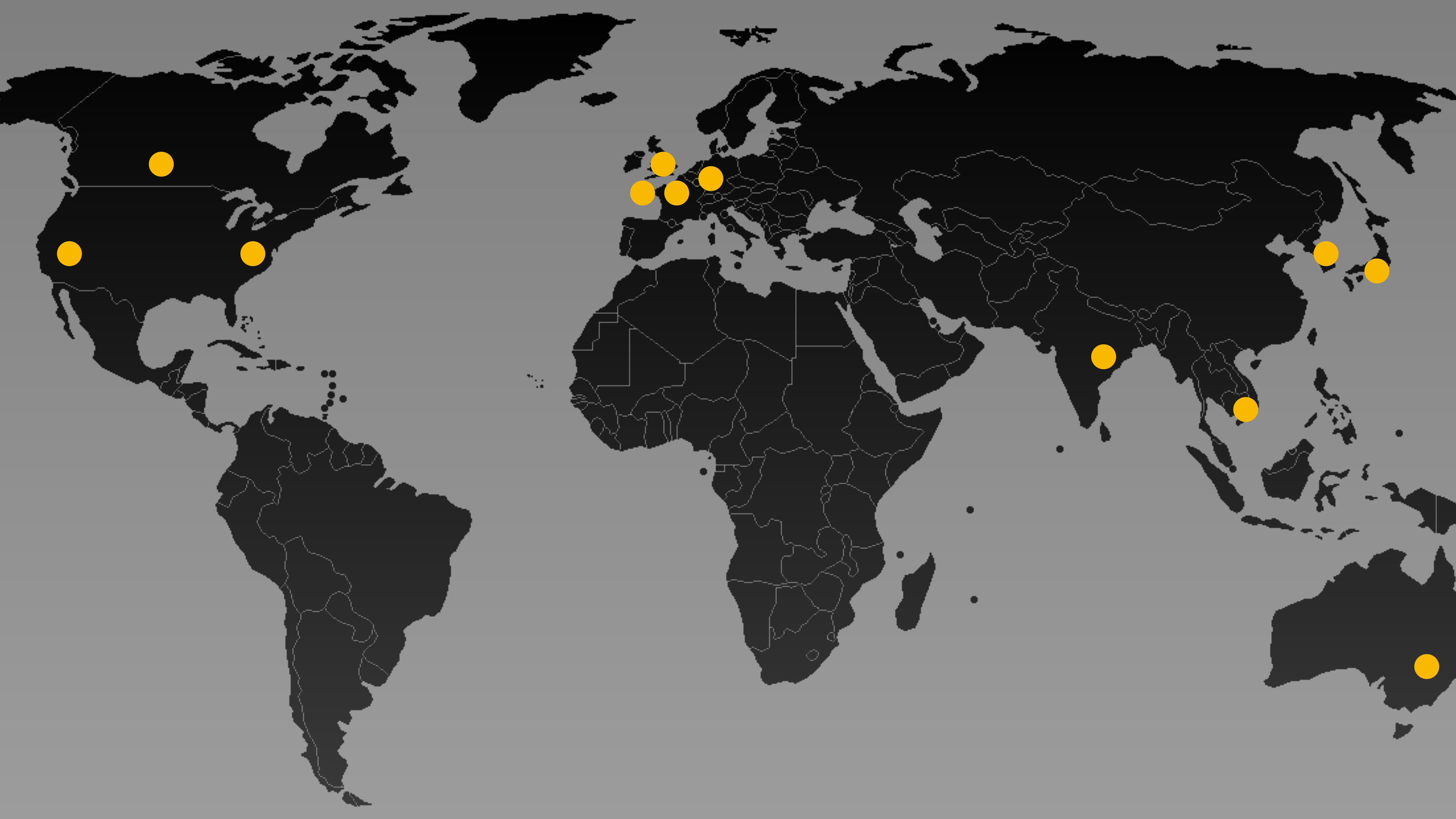
3f+1

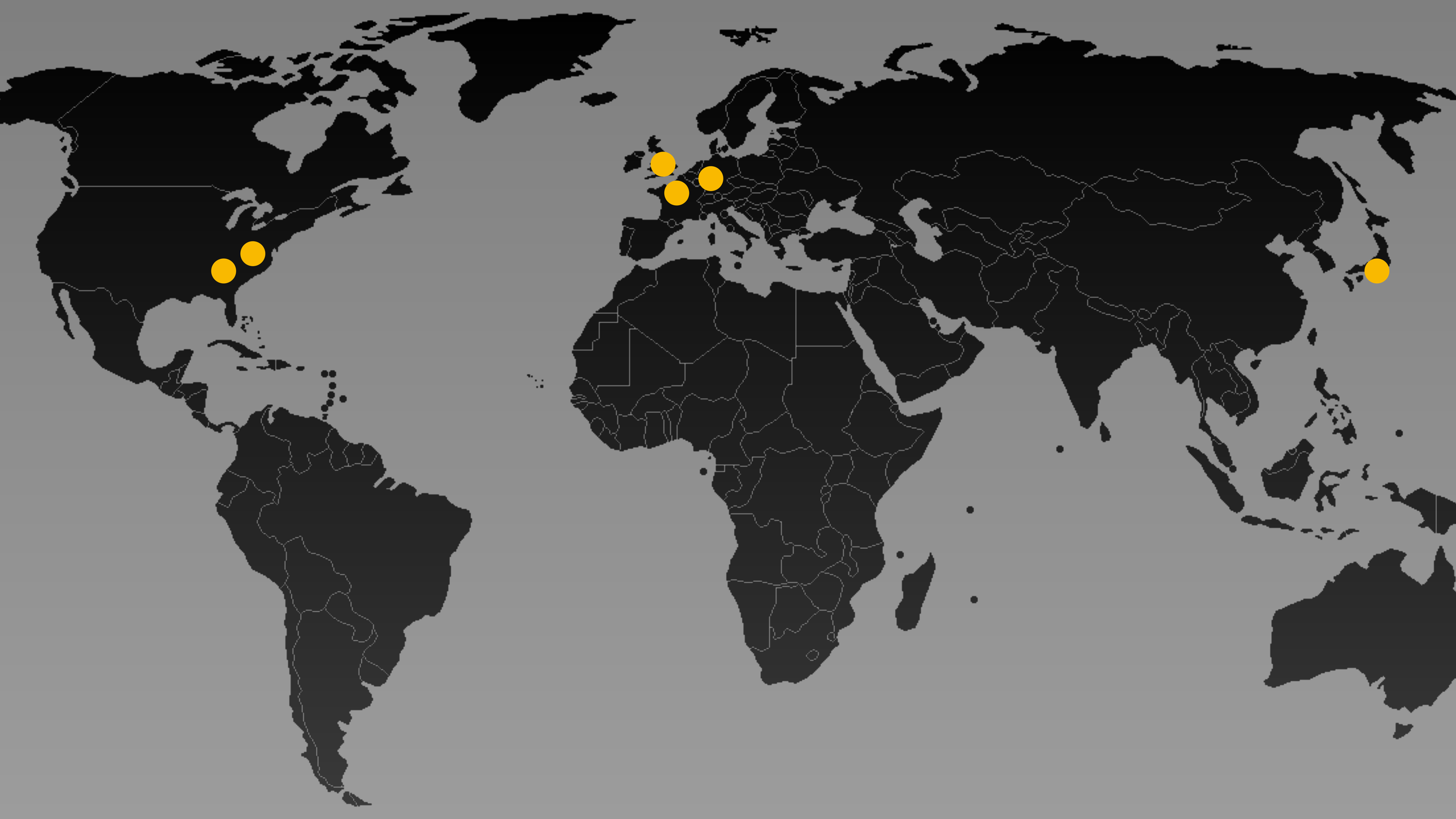
More rounds to commit (3) • Smaller quorums (2f+1) • Balanced safety-liveness

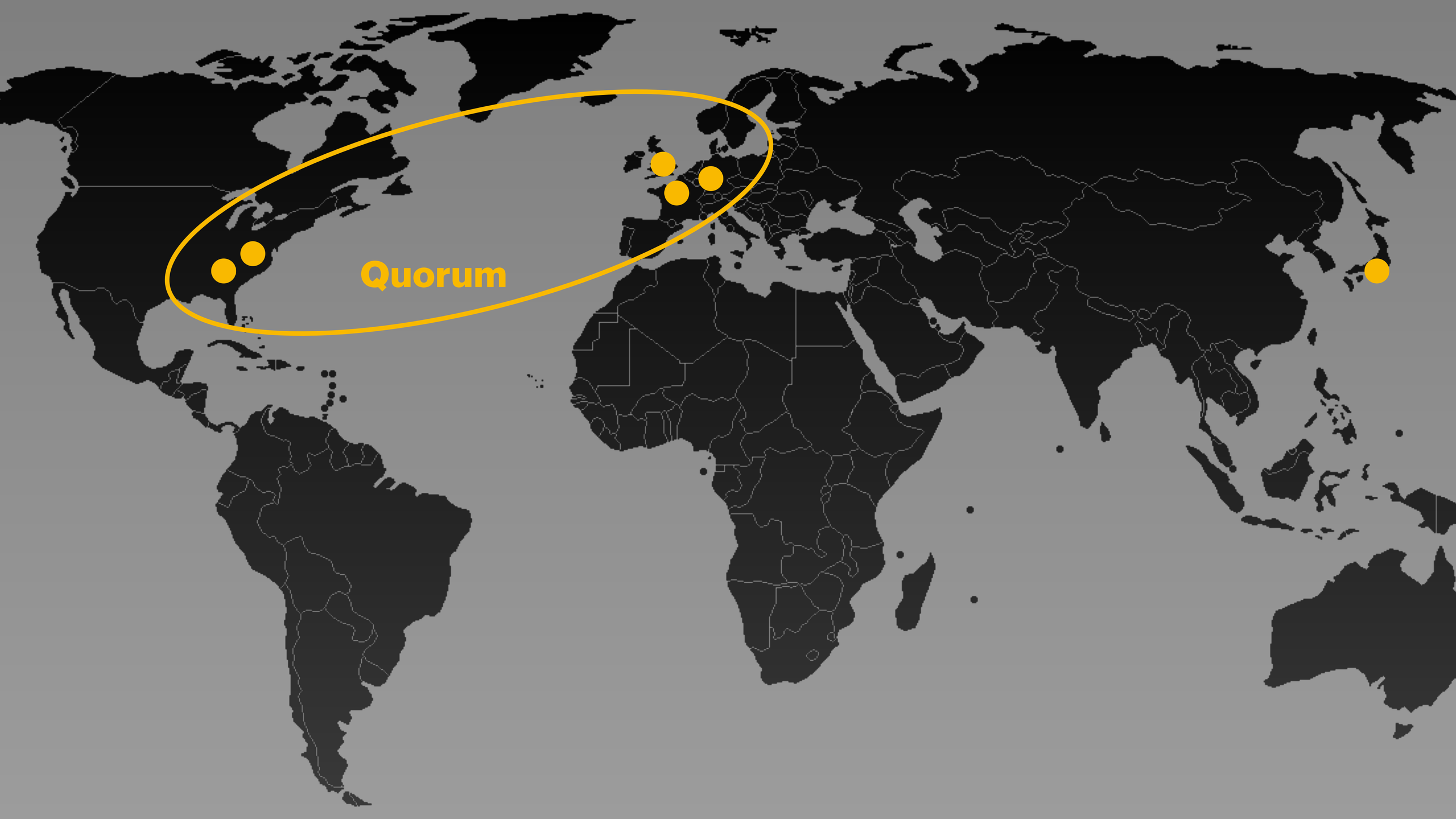
5f+1

Fewer rounds to commit (2) • Larger quorums (4f+1) • Lower liveness

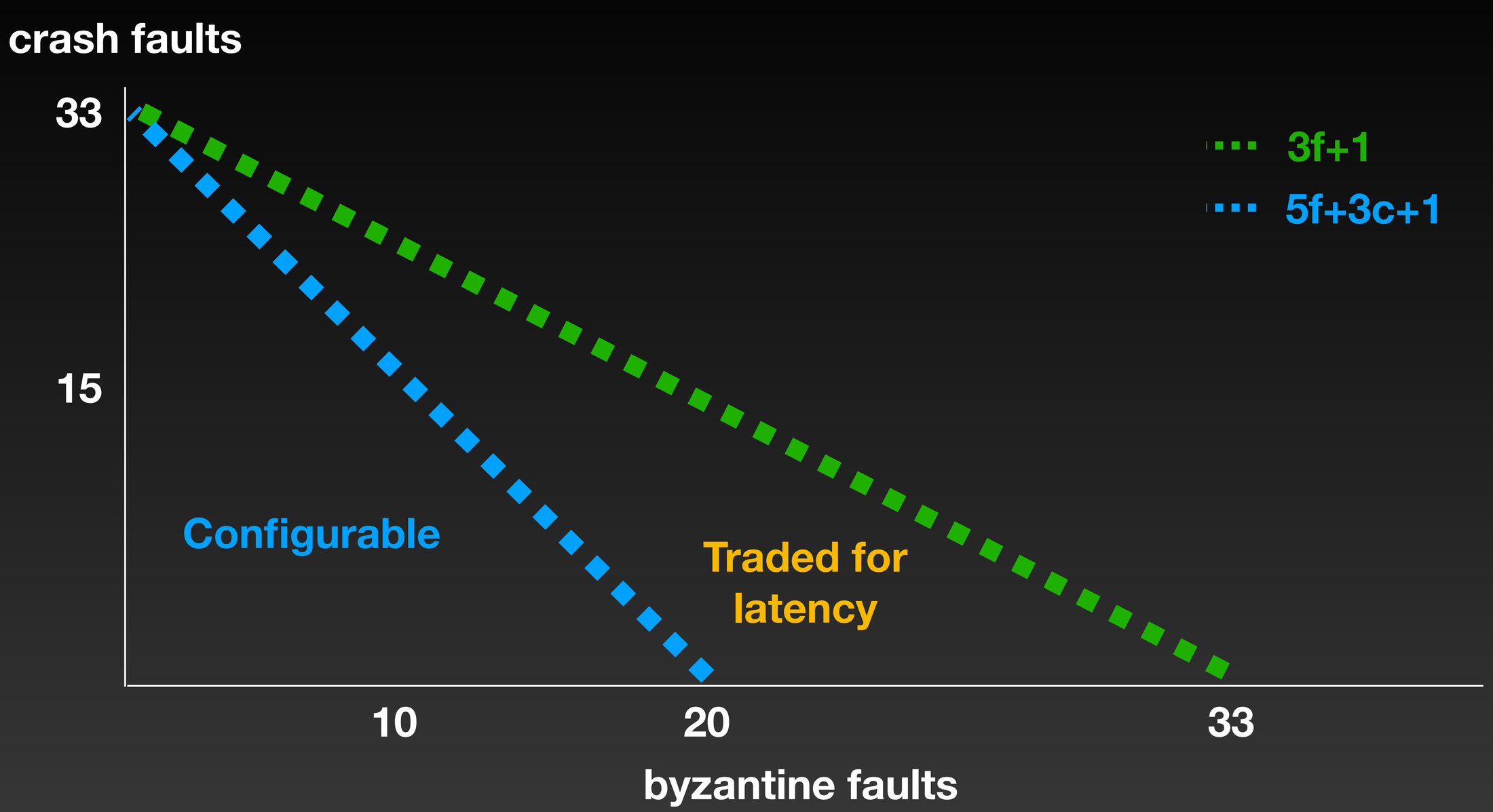

Load-induced
Block-Inclusion
Consensus
Execution

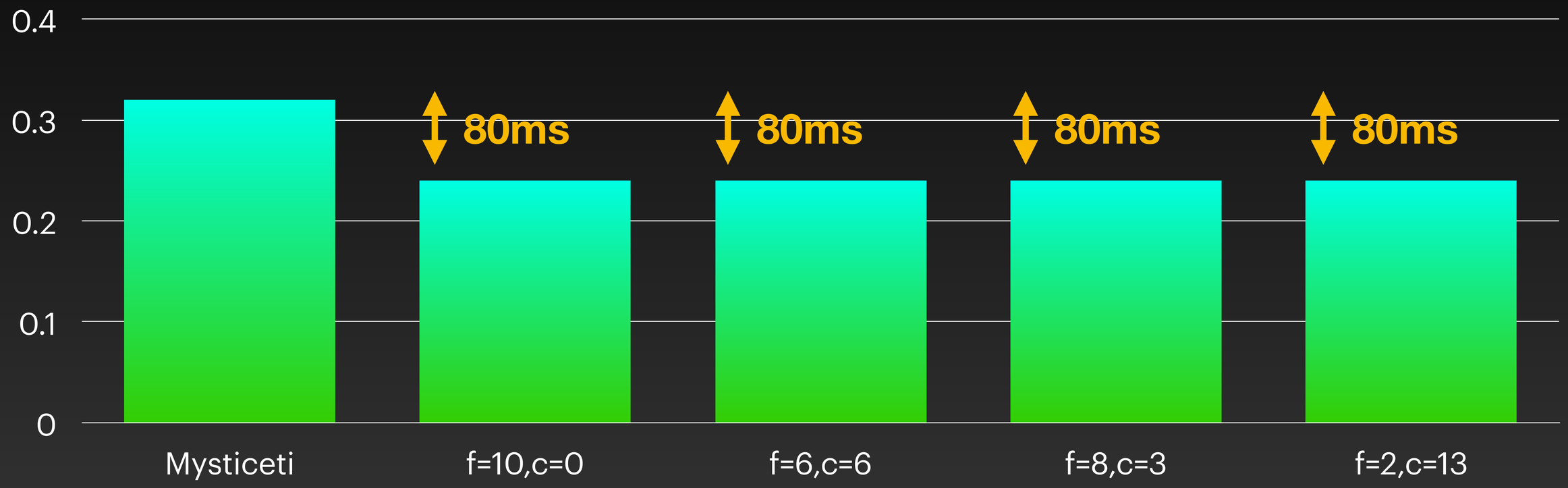


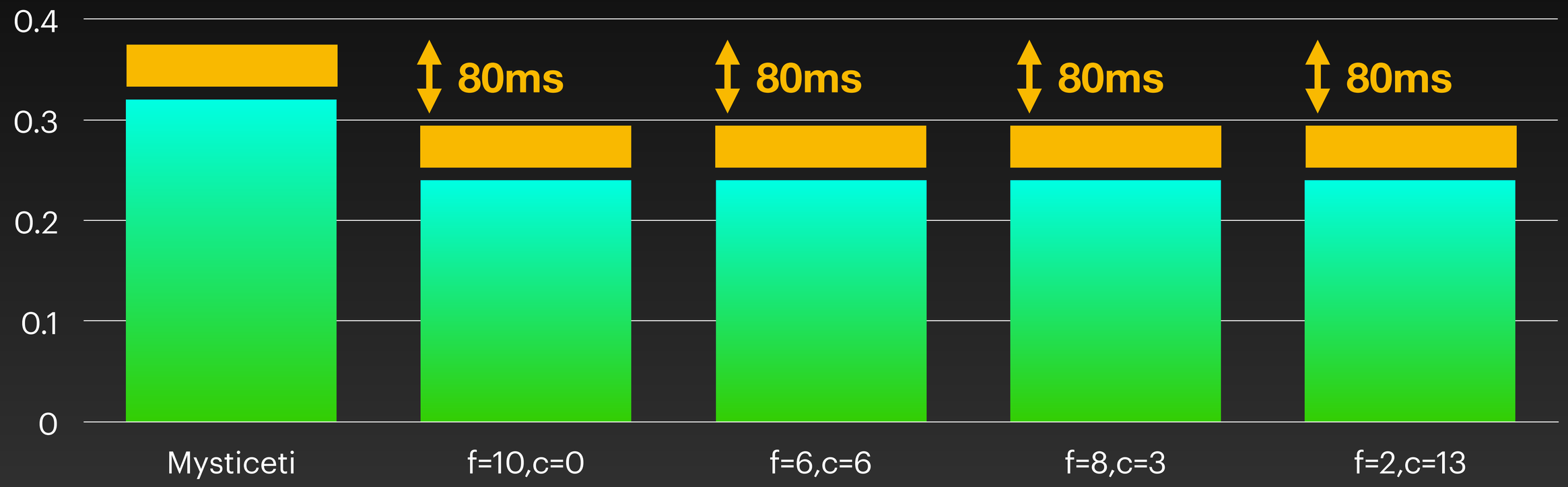



Quorum

5f+3c+1

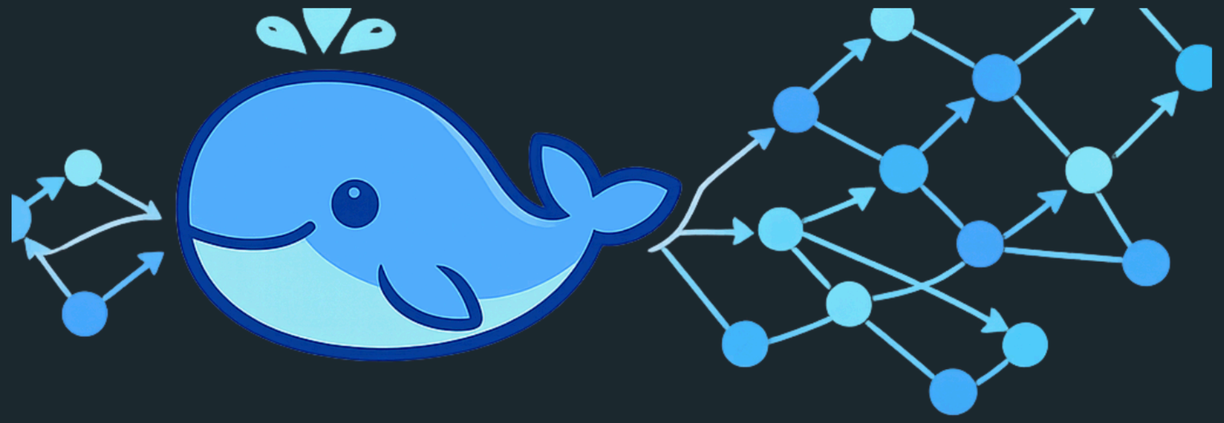

crash faults
33
15
3f+1
5f+3c+1
Configurable
Traded for
latency
10
20
33
byzantine faults


10k tx/s
0.4
0.3
0.2
0.1
0
80ms
80ms
80ms
80ms
Mysticeti
f=10,c=0
f=6,c=6
f=8,c=3
f=2,c=13


100k tx/s
0.4
0.3
0.2
0.1
0
80ms
80ms
80ms
80ms
Mysticeti
f=10,c=0
f=6,c=6
f=8,c=3
f=2,c=13

Mysticeti V3

f=?,c=?



Uncertified DAG Consensus

build passing | rustc 1.92+ | license Apache

This repository provides reference implementations of several uncertified DAG-based consensus protocols. The codebase is designed to be small, modular, and easy to benchmark and modify. It is not intended for production use, but relies on real networking, cryptography, and a persistent storage. It also ships with a discrete-event simulator for exercising the protocols under custom network and failure scenarios.

This repository currently supports Mysticeti, Mahi-Mahi, Blue Bottle, Cordial Miners (both the partially synchronous and asynchronous variants), and Nemo-Nemo.

Quick Start

https://github.com/asonnino/mysticeti

```rust
    /// "Mysticeti: Reaching the Latency Limits with Uncertified DAGs" NDSS 2025.
    /// <https://sonnino.com/papers/mysticeti.pdf>
    pub fn mysticeti(total_stake: Stake, leader_count: NonZeroUsize) -> Self {
        let quorum = 2 * total_stake / 3 + 1;
        Self {
            direct_commit_quorum: quorum,
            direct_skip_quorum: quorum,
            anchor_link_size: 1,
            wave_length: 3,
            leader_count,
            pipeline: true,
            leader_wait: true,
            require_crypto: true,
        }
    }

    /// Blue Bottle
    ///
    /// "BlueBottle: Fast and Robust Blockchains through Subsystem Specialization"
    /// <https://sonnino.com/papers/bluebottle.pdf>
    pub fn blue_bottle(total_stake: Stake, leader_count: NonZeroUsize) -> Self {
        let strong_quorum = 4 * total_stake / 5 + 1;
        let weak_quorum = 2 * total_stake / 5 + 1;
        Self {
            direct_commit_quorum: strong_quorum,
            direct_skip_quorum: strong_quorum,
            anchor_link_size: weak_quorum,
            wave_length: 2,
            leader_count,
            pipeline: true,
            leader_wait: true,
            require_crypto: true,
        }
    }

    /// Nemo-Nemo
    ///
    /// "Finding Nemo-Nemo: CFT DAG-based Consensus in the WAN"
    /// <https://sonnino.com/papers/nemo-nemo.pdf>
    pub fn nemo_nemo(total_stake: Stake, leader_count: NonZeroUsize) -> Self {
        let quorum = total_stake / 2 + 1;
        Self {
            direct_commit_quorum: quorum,
```

We are getting limited returns on algorithmic improvements of BFT consensus

controversial

High resource utilisation: throughput

Protocols commit often: latency

Engineering caught up with research

Change the assumptions (vs classic 3f+1)

Specialised chains (vs one chain to rule them all)

Other components are bottleneck (network, execution, etc)

Block Synchroniser

Theory
Reliable Broadcast • Proactive dissemination

Prototype
Optimistic dissemination • Random pull on a timer (if needed)

Production
Optimistic dissemination • Proactive re-sharing on timer • Bulk-sync of past commits

Transactions Duplication

DAG-based protocols

No deduplication until the first dissemination is complete

Multi-proposer protocols

No time to load and deserialise transactions for deduplication

Heavily pipelined protocols

No time time to interpret previous leader’s transactions for semantic deduplication

Why DAGs

Narwhal (sections 1-4) • https://sonnino.com/papers/narwhal-and-tusk.pdf

Robustness & Throughput

Bullshark (partially synchronous) • https://sonnino.com/papers/bullshark-simple.pdf

Latency

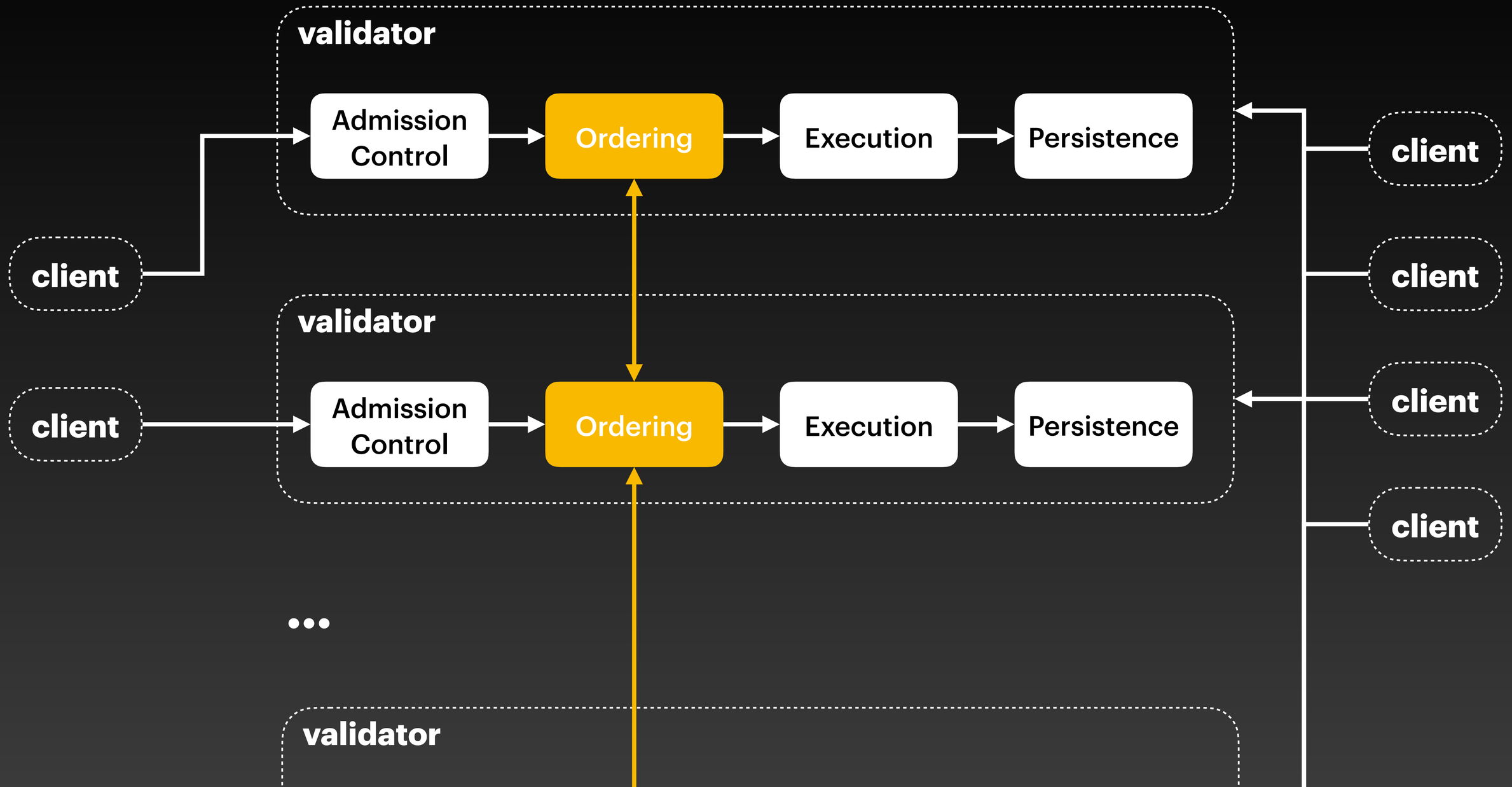
Shoal++ • https://arxiv.org/pdf/2405.20488v1

Uncertified DAG & Real World

Mysticeti (skip section 5) • https://sonnino.com/papers/mysticeti.pdf

Unstructured DAG

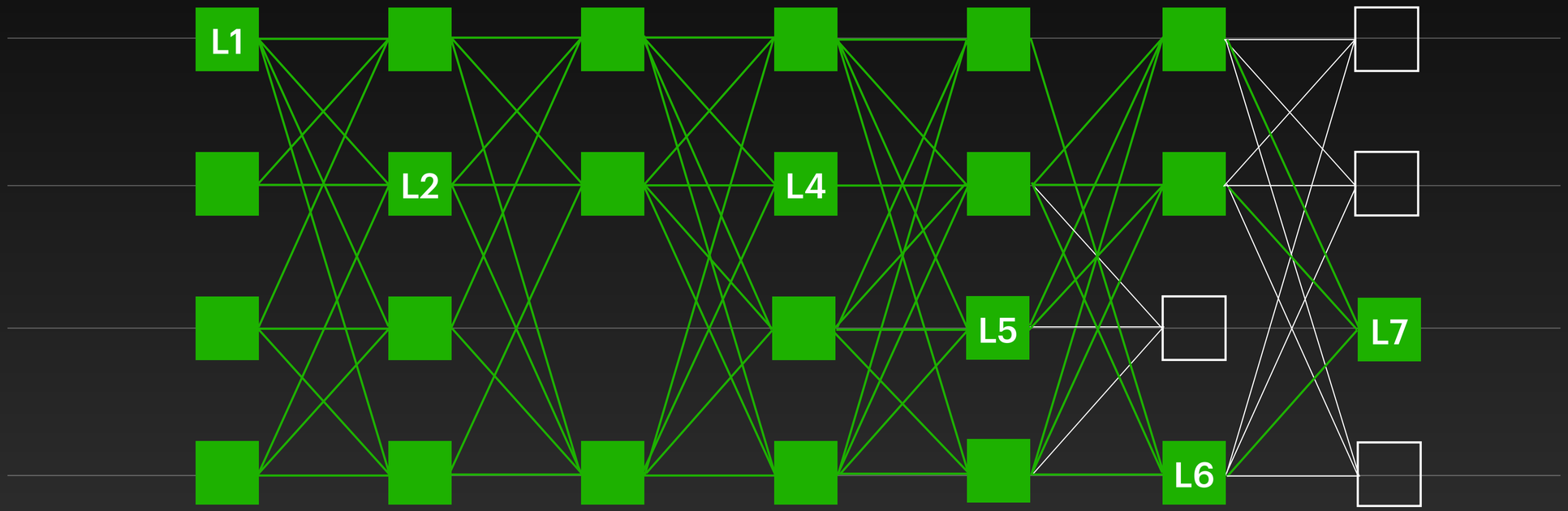
Autobahn • https://arxiv.org/pdf/2401.10369


validator
Admission Control
Ordering
Execution
Persistence
client
validator
Admission Control
Ordering
Execution
Persistence
client
validator
client
client
client
client
...

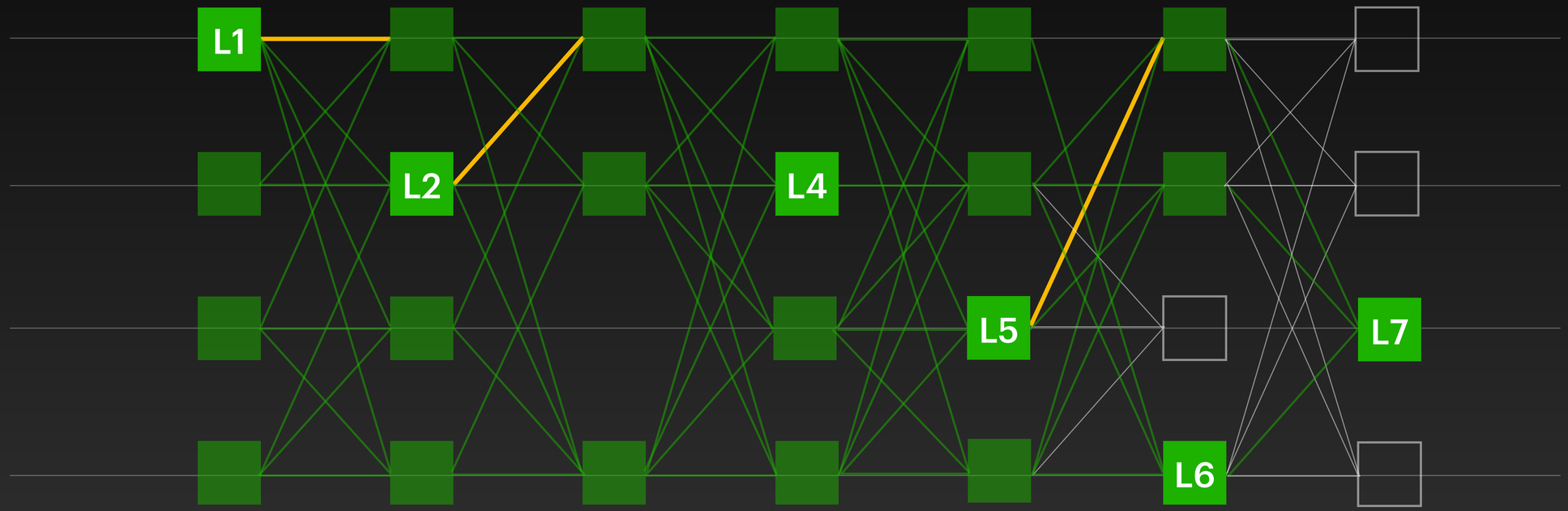
alberto@mystenlabs.com

EXTRA:
Leader Reputation

Past Commits

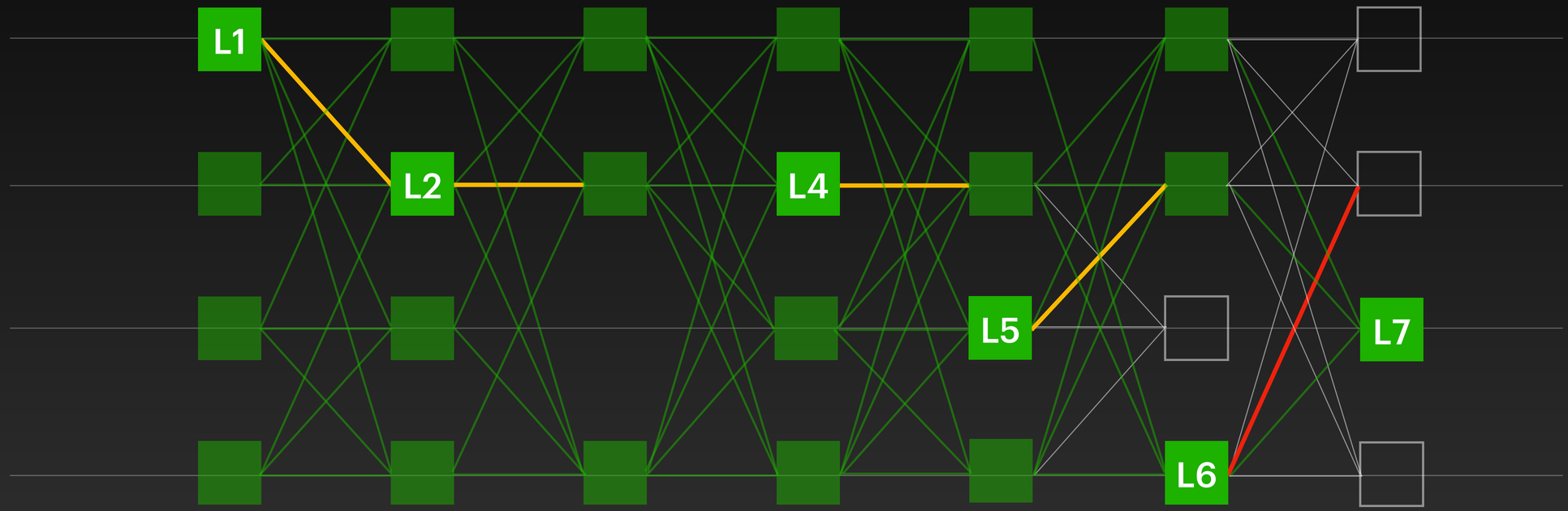


Compute Reputation Scores



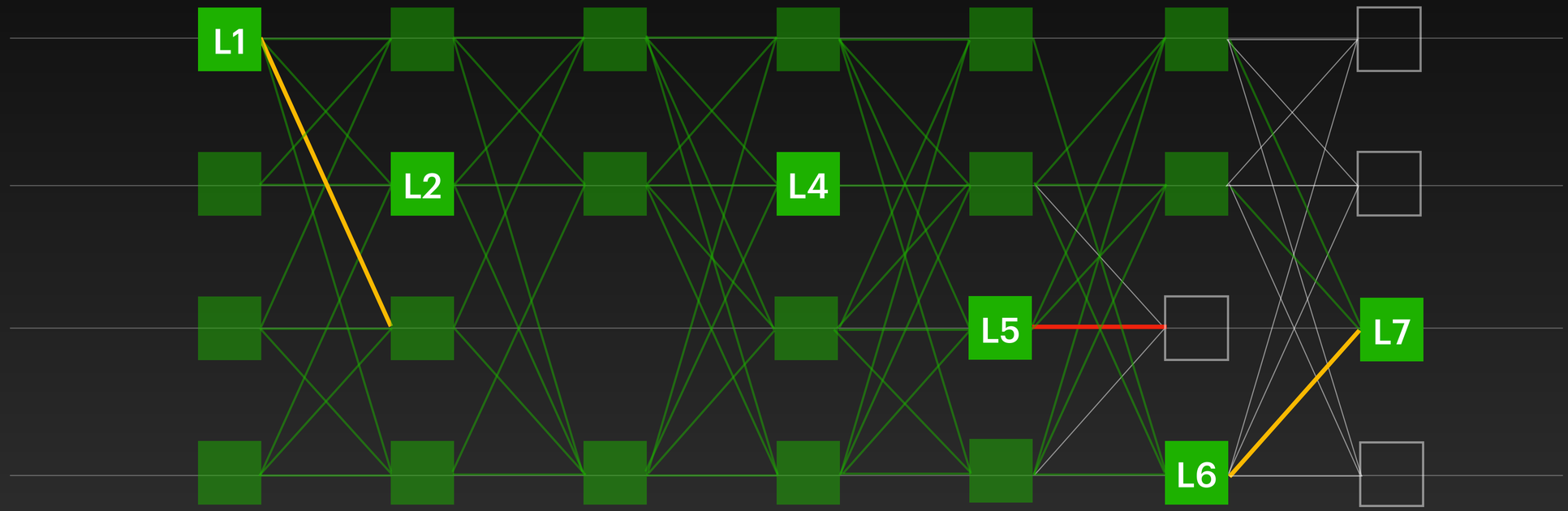
node 1: 3

Compute Reputation Scores



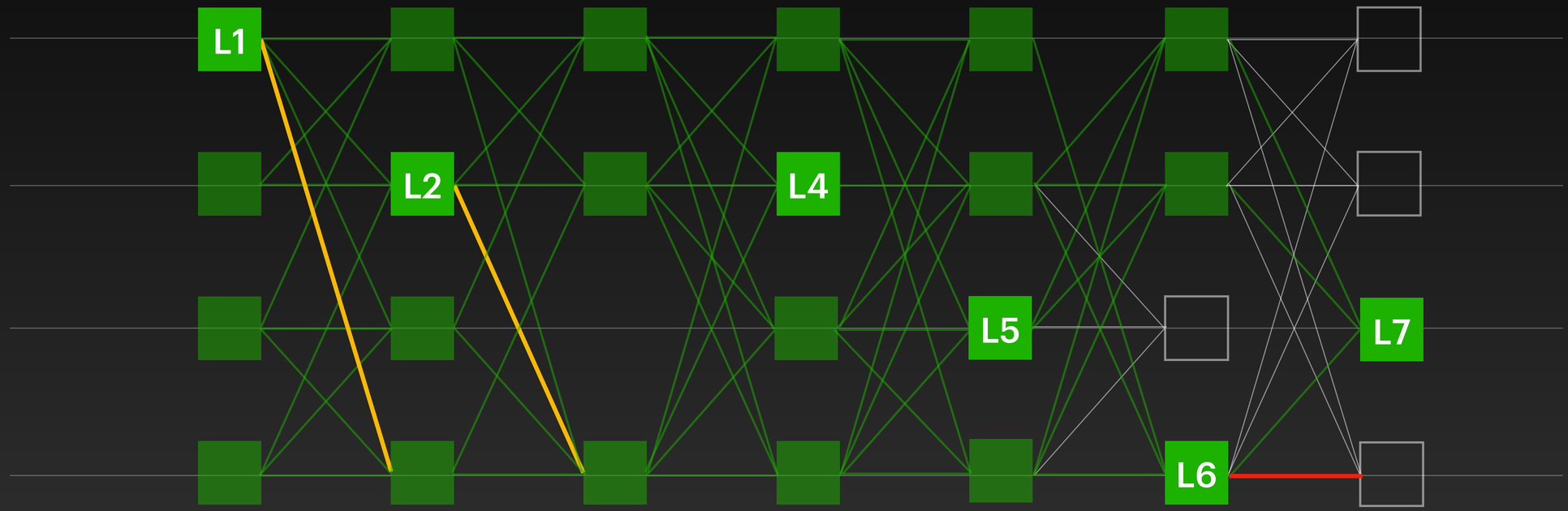
node 1: 3 **node 2:** 4

Compute Reputation Scores



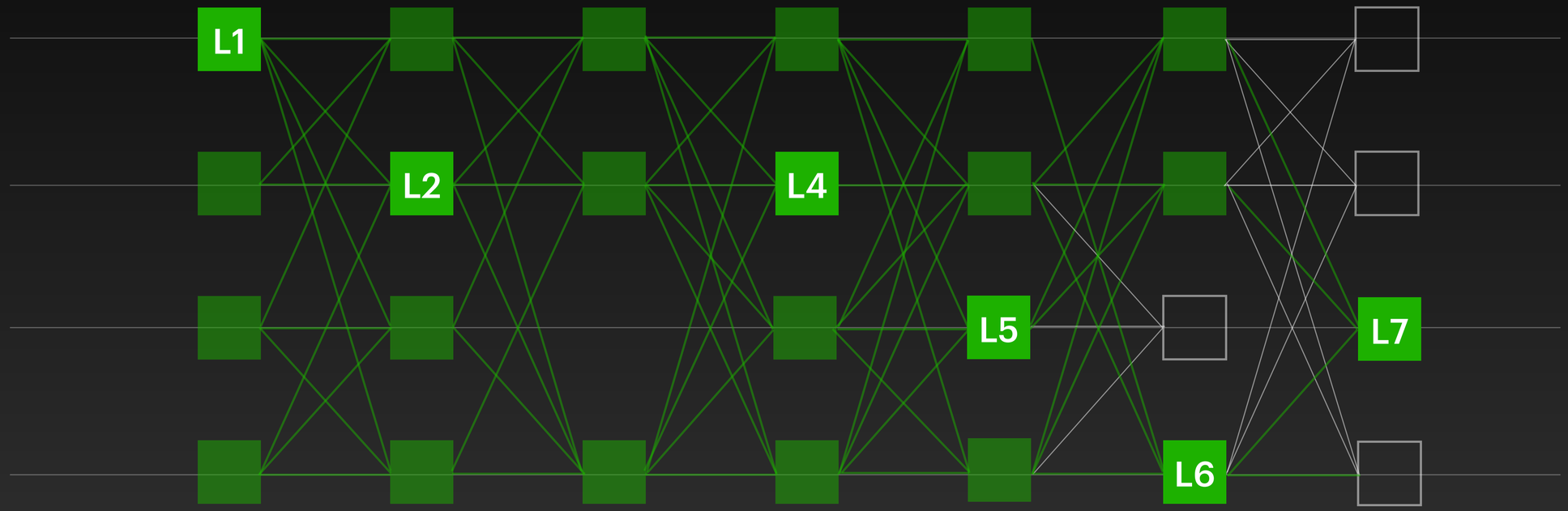
node 1: 3 node 2: 4 node 3: 2

Compute Reputation Scores



node 1: 3 node 2: 4 node 3: 2 node 4: 2

Future Leaders



node 1: 3 node 2: 4 node 3: 2 node 4: 2