

# Shoal: Improving DAG-BFT Latency and Robustness

FC'24

Alexander Spiegelman, Rati Gelashvili, Balaji Arun, Zekun Li, Aptos

## Contents

1	Context: DAG-based BFT Consensus . . . . .	4
1.a	BFT Consensus . . . . .	5
1.b	DAG application . . . . .	6
1.c	DAG Example . . . . .	7
1.d	Vertices dissemination . . . . .	8
1.e	Consensus Mechanism . . . . .	9
1.f	Consensus Structure . . . . .	10
1.g	DAG Waves example . . . . .	11
2	Problem . . . . .	12
2.a	First Slide . . . . .	13
3	Solution: Pipelining . . . . .	13

## Contents (ii)

3.a First Slide . . . . .	14
4 Evaluation . . . . .	14
4.a First Slide . . . . .	14

# 1 Context: DAG-based BFT Consensus

# BFT Consensus



- $N \geq 3f+1$  validators in total
- At most  $f$  validators are faulty

Global agreement on an infinitely growing sequence of some values.

# DAG application

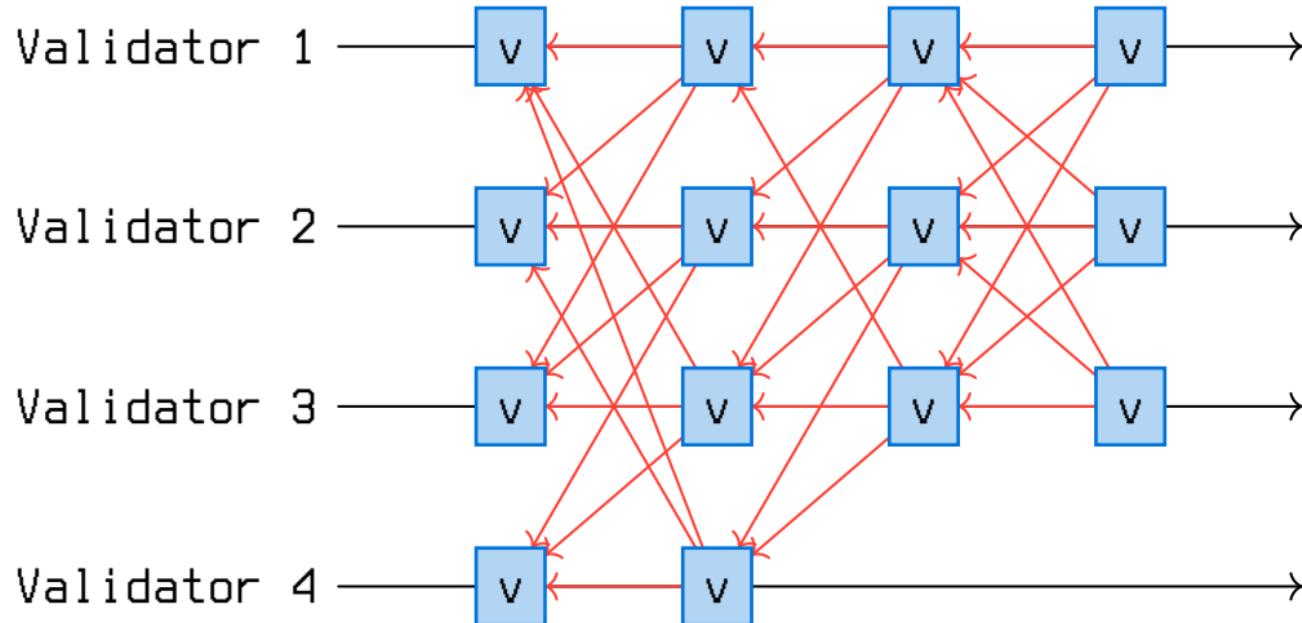


## Idea

Separate the network communication layer from the consensus logic.

- Each message contains a set of transactions, and a set of references to previous messages.
- Together, all the messages form a DAG that keeps growing - a message is a vertex and its references are edges.

# DAG Example



# Vertices dissemination



Unifies abstraction

Reliable BFT broadcast (Not all protocols)

Result:

- All honest validators eventually deliver the same vertices and all vertices by honest validators are eventually delivered.
- Causal history of any vertex in both local views is exactly the same.

## Consensus Mechanism



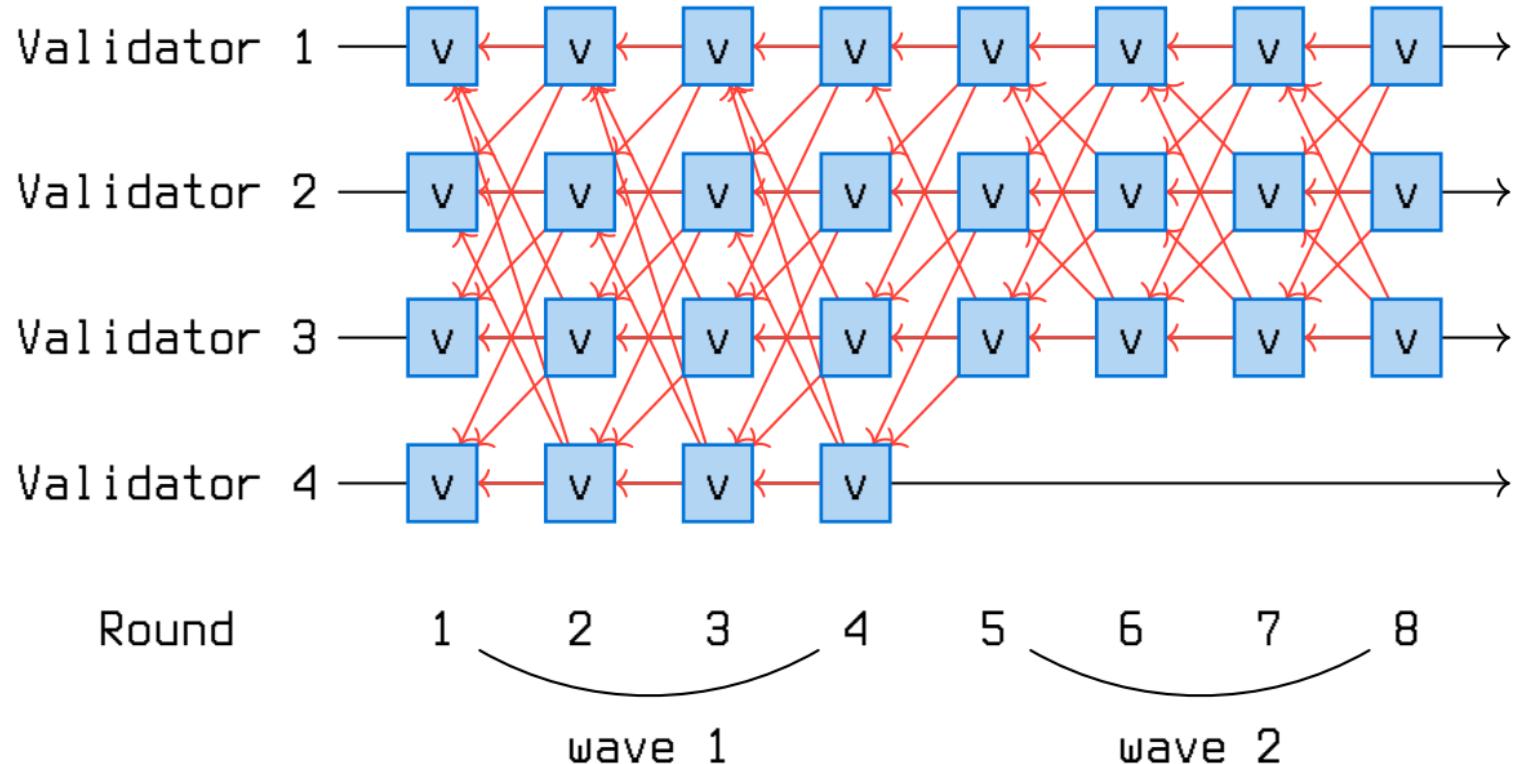
- Solving locally!!
- No need of any extra communication.

# Consensus Structure



- Consensus divided into rounds
- Rounds groups waves
- Each wave contains a leader

# DAG Waves example



## 2 Problem

# 3 Solution: Pipelining

# 4 Evaluation