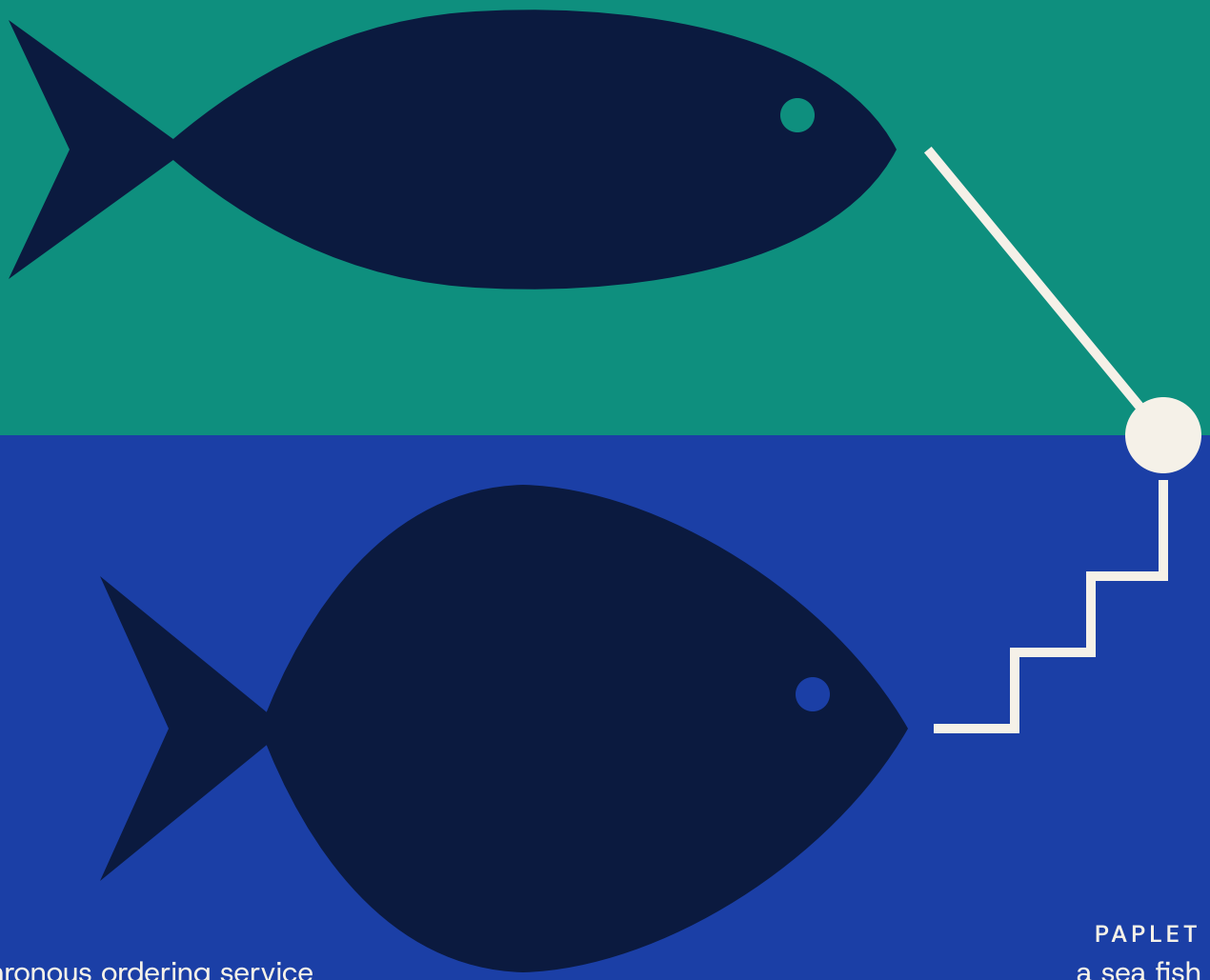
RIVER
the fast path

ROHU
a river fish



OCEAN
the asynchronous ordering service

PAPLET
a sea fish

Rohu-Paplet

Fast finality that survives recovery.

Rohu-Paplet is a BFT consensus design with a fast path for each block, kept beside an asynchronous DAG that only orders recovery.

NAMAN BAJPAI

NOVEMBER 2026

A timeout does not erase votes.

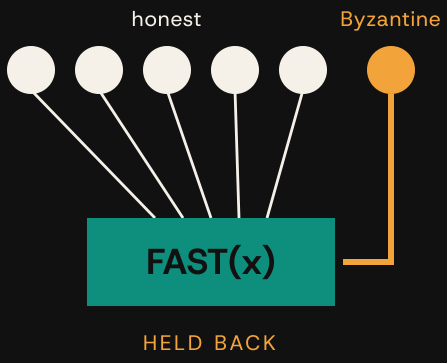
Fast consensus protocols finalize a block in two message delays when the network behaves. When it does not, a timer fires and recovery takes over.

Validators may already have signed enough PREPAREs to form a fast certificate. A Byzantine collector can hold that certificate back and show it after recovery picks something else.

THE LATE CERTIFICATE

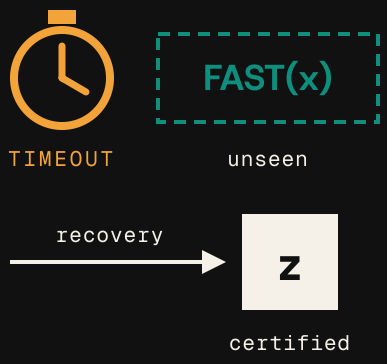
f = 1 · n = 6 · FAST needs 4f+1 = 5 PREPAREs

1 Votes go out



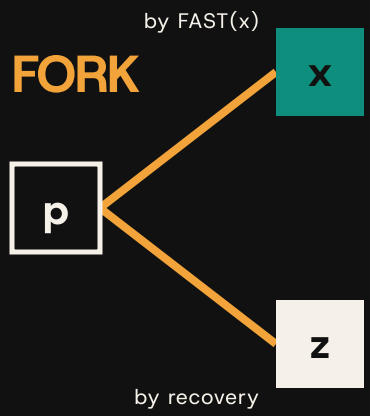
Five validators sign PREPARE(x). That makes FAST(x). A Byzantine collector holds it back.

2 The timer fires



The timer fires. A careless recovery sees no FAST and certifies another child, z.

3 The late reveal



FAST(x) appears. Block p now has two certified children, x and z.

Recovery cannot read only what its participants have seen. It has to read what could already finalize.

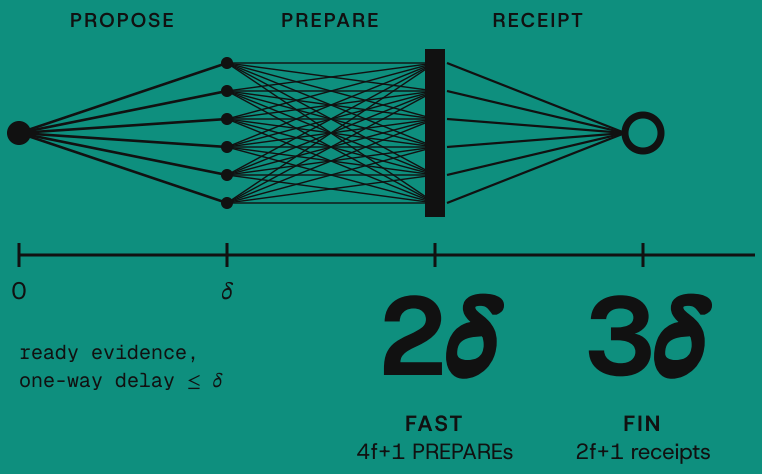
Two routes, one order.

Rohu-Paplet runs both routes all the time. River tries to finalize the next block directly. Ocean keeps ordering control records in the background, whatever River is doing.

RIVER · THE FAST PATH

Each certified block p gets exactly one fast attempt at its child. The leader proposes a child x with all its evidence. Each open validator saves its choice to disk, then sends one PREPARE.

A validator that holds a complete U, 3f+1 PREPAREs for x, can also send a receipt.



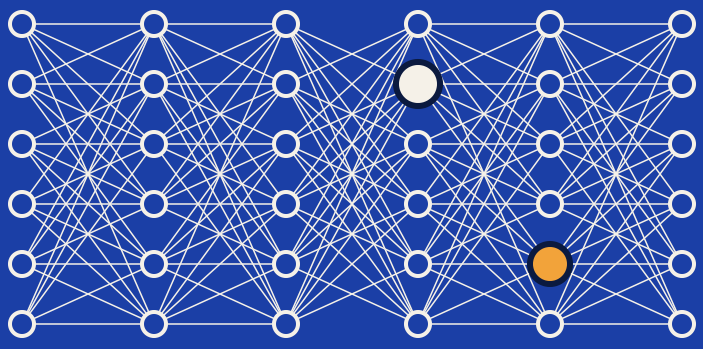
OCEAN · THE ASYNCHRONOUS ORDERING SERVICE

Ocean never stops.

Ocean is a DAG of control vertices, built round by round with reliable broadcast. Each vertex needs at least 4f+1 parents from the previous round.

Rounds advance even when vertices carry no transactions, so recovery never waits on data.

Ocean supplies accepted cuts and orders recovery records for blocks that stall. Neither job lets Ocean choose the next block by itself.



leader vertex, elected by a common coin

vertex that carries a resolution record

< 5/4 expected waves until the coin hits the common core, under an ideal coin. A count of waves, not seconds.



Both routes authorize the same kind of object: an exact child of a certified parent. The application never asks which route supplied the proof.

Recovery keeps what could finalize.

ONE COUNTING ARGUMENT

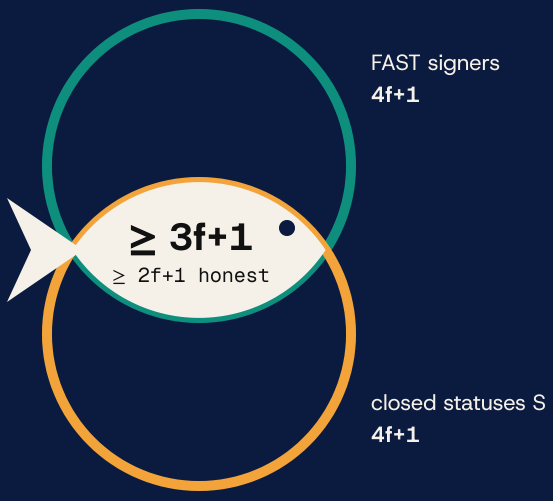
$|A \cap B| \geq |A| + |B| - n$

`3f+1, 3f+1` share at least f+1, so one is honest

`4f+1, 3f+1` share at least 2f+1

`4f+1, 4f+1` share at least 3f+1, so 2f+1 are honest

Here n = 5f+1, and at most f validators misbehave. An honest validator never signs two conflicting values for the same block.



CLOSE, THEN REPORT

If the child of p does not finalize in time, a local timer closes the domain of p. A closed validator never signs another PREPARE or receipt for a child of p.

Its closed status carries its own PREPARE, if it sent one, and every complete U it kept. Seeing f+1 closed statuses makes a validator close too.

protect(S) S = 4f+1 closed statuses

1	X	if S contains a complete U for x
2	X	else, if at least 2f+1 statuses report their own PREPARE for x
3	⊥	otherwise, this set forces no child

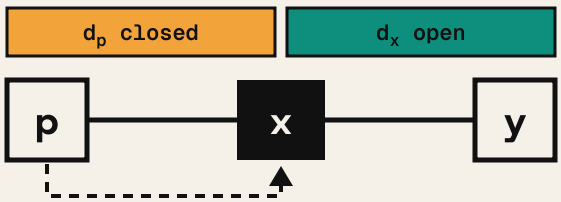
The earliest valid resolution for p in Ocean order certifies its child. Later records for p change nothing.

WHY A HIDDEN FAST SURVIVES

Suppose a FAST for x exists, but nobody in recovery has seen it. Its 4f+1 signers and the 4f+1 status authors share at least 3f+1 validators, so at least 2f+1 honest ones.

Each signed its PREPARE for x before it closed. So line 2 selects x. A hidden FIN works the same way: some honest receipt signer is in S and shows its U.

ONE DOMAIN PER BLOCK



late resolution of p: only x again

Closing the domain of p never touches the domain of x. There is no global stop and no checkpoint ceremony.

Two fewer validators.

The 5f+1 comes from the recovery rule, not from the problem. Today's rule counts reports and ignores who wrote them. A rule that reads authors needs fewer validators.

$$2q_F + q_S \geq 2n + 2f + c$$

q_F fast quorum · q_S closed statuses · both $\leq n - f$

5f−1

c = −1

Drop a leader caught signing two children. Wait for one more status.

PROVEN FLOOR FOR TWO-STEP BFT

5f

c = 0

Read the leader's signature on each PREPARE. No extra wait.

5f+1

c = +1

Count reports. Ignore who wrote them.

ROHU-PAPLET TODAY

At f = 20 that is 99, 100, or 101 validators, with the same 2δ fast path. Kuznetsov, Tonkikh, and Zhang proved the 5f−1 floor and gave the exclusion step.

MESSAGES PER OCEAN ROUND · SAME N · NO FAULTS

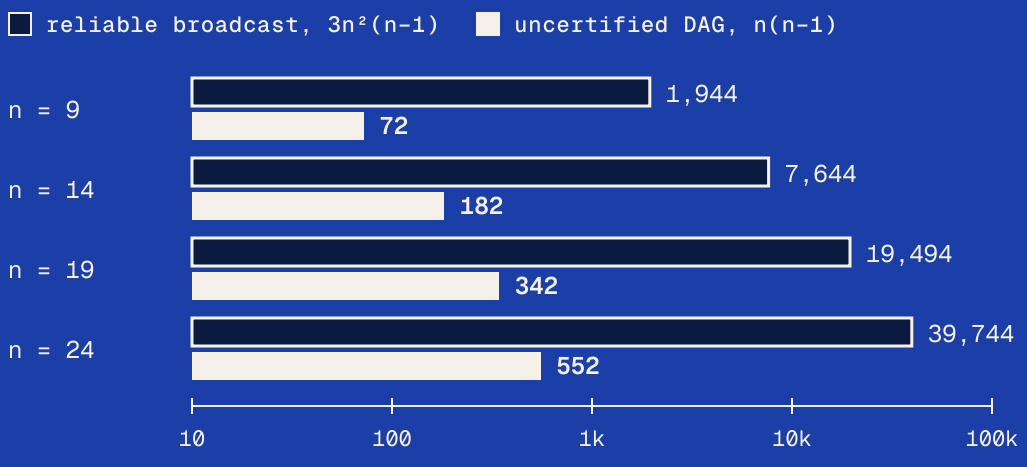
One push per vertex.

3n×

fewer messages per round

The 5f−1 variant also drops reliable broadcast. Ocean runs on an uncertified DAG, where each vertex goes once to each validator. Its commit rule is the asynchronous rule of Cordial Miners.

With reliable broadcast, every vertex is echoed and relayed. The benchmark matched 3n²(n−1) exactly at n = 6 to 21. A simulation of n(n−1) adds 4 to 9 percent for fetches.



5δ

A good-case Ocean commit takes about 5δ, not 4δ. The price is speed.

Each piece exists somewhere already.

BlueBottle

Runs an asynchronous uncertified DAG at 5f+1.

Hydrozoan

Runs a 5f−1 fast path on a DAG. It is live only under partial synchrony.

FinWhale

Runs a 5f−1 fast path and its recovery inside a DAG, under partial synchrony.

Alpenglow

Finalizes in one round of votes with 80% of stake, or two rounds with 60%, under partial synchrony.

Cordial Miners

Gives the asynchronous commit rule that the 5f−1 Ocean uses, with five-round waves.

Kuznetsov, Tonkikh, Zhang

Proved that 5f−1 is the floor for two-step consensus. The exclusion step is theirs.

We found no published protocol with a two-delay fast path at 5f−1 that also stays live under full asynchrony.

Rohu-Paplet at 5f−1 has both on paper.

STATUS

Hand proofs

for the recovery bounds and for the Ocean rule.

f ≤ 5

Exhaustive search found no status set that loses a late FAST.

f ≤ 3

Adversarial fuzzing of the 5f−1 Ocean rule.

Not yet

Not mechanized. Not peer reviewed.

→ singularitybft.vercel.app
→ thesingularity.so
→ thesingularity.sh

Naman Bajpai

NOVEMBER 2026