Fast Consensus in Weakly Byzantine Asynchronous Environments

binary consensus

asynchronous systems

weakly byzantine environments

weakly / strongly one-step protocols

asymmetric consensus

Aleksander Kampa
alex.kampa@aragon.org

ARAGON
ZK RESEARCH

TL;DR

Under certain realistic conditions, asynchronous binary consensus can be reached **in a single communication** step by slightly weakening fault tolerance assumptions.

For example, a network of 13 nodes can tolerate 4 faulty nodes, of which 3 can be Byzantine, and still often achieve one-step consensus.

ARAGON
ZK RESEARCH

Efficient Consensus in Distributed Systems

Motivation:

- Quickly reaching consensus in distributed systems is crucial

Observations:

- Crashes and Byzantine behaviour are rare
- Network performance is generally reliable and fast
- Weaker fault-tolerance assumptions can be acceptable

Our objective:

- Achieve one-step consensus when conditions are favorable
- Fallback to a generic protocol when necessary

ARAGON
ZK RESEARCH

How?

(1) Reduce fault tolerance assumptions

(2) Introduce asymmetric consensus

(3) Wait for more messages than standard protocols

ARAGON
ZK RESEARCH

The AWB Setting

AWB = Asynchronous Weakly Byzantine environment

ARAGON
ZK RESEARCH

The setting (1)

We are in the context of a classic problem: reaching consensus in a distributed system

There are n nodes, each with an initial value of 0 or 1.

The nodes aim to reach consensus on a value by following a randomised protocol.

The nodes communicate over a reliable asynchronous network. All messages are eventually delivered, but there is no upper bound on message delays.

The setting (2)

The nodes that follow the protocol exactly are said to be correct.

At most t of the n nodes are faulty. Among the potentially faulty nodes, at most t’ ≤ t can be Byzantine, the remaining being crash-prone.

When t' = t, this is the classic Byzantine environment.

When t' < t, we call this a **weakly Byzantine environment**.

ARAGON
ZK RESEARCH

The setting (3)

The adversary:

- has complete control over the Byzantine nodes
- can make crash-prone nodes crash at any time
- can see the content of all messages
- can decide on the order in which messages are delivered and can delay message delivery

However, the adversary cannot impersonate nodes.

ARAGON
ZK RESEARCH

The setting (4)

We require the usual conditions from our protocol:

- *agreement*: all correct nodes decide the same (consensus) value
- *validity*: the consensus value is one proposed by a correct node
- *termination*: all correct nodes decide with probability 1

We also require *round finality*: once a node decides on a value, that decision is irreversible. In a (fully) Byzantine setting it is well known that consensus is then only possible if strictly less than n/3 nodes are Byzantine.

ARAGON
ZK RESEARCH

Binary Consensus in AWB

ARAGON
ZK RESEARCH

Three useful concepts

- **Zugzwang**: At any stage of a consensus protocol, a node may have to make a decision after receiving only n - t messages.

- **Correct Value condition**: After receiving n-t messages, a node will know at least one value that has been sent by a honest node. Given that a node is only guaranteed to receive (n-t)/2 values for any one value, this constraint can be expressed as (n-t)/2 > t' which implies the following condition:

$$n > t + 2t'$$

- **Commitment threshold (for n-t messages)**: If a node receives n-t identical messages for a value b, no other node can receive n-t messages with a different value ¬b. This requires:

$$n > 2t + t'$$

ARAGON ZK RESEARCH

Binary Consensus in AWB

Can we improve on the well-known n > 3t condition? We can!

We take an existing binary consensus protocol* and find that it relies on two essential points:

- Correct Value condition, which requires n > t + 2t'
- Commitment threshold for n-t, which requires n > 2t + t'

As t ≥ t', this implies that:

Binary AWB consensus is possible if n > 2t + t'

* *A Simple and Efficient Asynchronous Randomized Binary Byzantine Consensus Algorithm*, Tyler Crain, 2020

Fast Consensus in Weakly Byzantine Asynchronous Environments

One-step Consensus

ARAGON
ZK RESEARCH

Definitions

- A one-step protocol allows a decision in one communication step under certain conditions.
- A protocol is ***strongly one-step**** if it allows a one-step decision when all correct nodes have the same initial value, even when some nodes are Byzantine.
- A protocol is ***weakly one-step**** it allows a one-step decision when all correct nodes have the same initial value and there are no Byzantine nodes.

* These terms were defined in «*Bosco: One-Step Byzantine Asynchronous Consensus*», Yee Jiun Song and Robbert van Renesse, 2008

ARAGON
ZK RESEARCH

Conditions for one-step consensus in AWB

In a weakly Byzantine environment, the conditions for achieving one-step consensus are:

- **Weakly one-step requires $n > 3t + 2t'$**
- **Strongly one-step requires $n > 3t + 4t'$**

This generalises results obtained by Song and van Renesse* in the usual Byzantine setting

* *Bosco: One-Step Byzantine Asynchronous Consensus*, Yee Jiun Song and Robbert van Renesse, 2008

ARAGON
ZK RESEARCH

The W-Bosco protocol

The W-Bosco for weakly Byzantine environments protocol is a slight modification of the Bosco protocol, introduced by Song and van Renesse*

	Input: v_p
1	Broadcast (**VOTE**, v_p) to all nodes
2	Wait until $n - t$ **VOTE** messages have been received
3	**if** $> \frac{n+t+2t'}{2}$ **VOTE** messages contain the same value v **then**
4	DECIDE(v)
5	**if** $> \frac{n-t}{2}$ **VOTE** messages contain the same value v, **then**
6	$v_p \leftarrow v$
7	**Underlying-Consensus**(v_p)

NB Setting t=t' in line 3 gives us the original Bosco protocol

* *Bosco: One-Step Byzantine Asynchronous Consensus*, Yee Jiun Song and Robbert van Renesse, 2008

ARAGON
ZK RESEARCH

Options for weakly one-step consensus with n=50

	t	t'	$3t+2t'$
Bosco	9	9	45
W-Bosco	10	9	48
	11	8	49
	12	6	48
	13	5	49
	14	3	48
	15	2	49
	16	0	48

W-Bosco does not improve on Bosco, but it introduces an element of choice: if less Byzantine failures can be tolerated, crash resilience can be increased.

Fast Consensus in Weakly Byzantine Asynchronous Environments

ARAGON
ZK RESEARCH

Asymmetric Consensus

ARAGON
ZK RESEARCH

Symmetric versus asymmetric protocols

A *symmetric* consensus protocol treats both input values in a similar way, without giving preference to one or the other.

However, it is sometimes known in advance which value is most likely to be the consensus value. For example, in a leader-based protocol, the value proposed by the round leader will usually be accepted.

We define an ***asymmetric consensus protocol*** as one that favours one of the two values.

ARAGON
ZK RESEARCH

Conditions for asymmetric one-step consensus in AWB

In a weakly Byzantine environment, the conditions for achieving asymmetric one-step consensus are:

- **Weakly one-step requires $n > 2t + 2t'$**
- **Strongly one-step requires $n > 2t + 3t'$**

Note that these constraints are more favourable than for symmetric consensus.

ARAGON
ZK RESEARCH

The W-Bosco-0 protocol

This protocol favours the value 0 in the first step.

```
     Input: v_p
1    Broadcast (VOTE, v_p) to all nodes
2    Wait until n − t VOTE messages have been received
3    if > t + 2t' VOTE messages contain the same value 0 then
4      DECIDE(0)
5    if > t' VOTE messages contain the value 0, then
6      v_p ← 0
7    else
8      v_p ← 1
9    Underlying-Consensus(v_p)
```

Note that in line 3, we require only t + 2t’ + 1 identical values to decide, compared to > (n + t + 2t’)/2 values in W-Bosco.

ARAGON
ZK RESEARCH

Options for weakly one-step consensus with n=50: Symmetric versus Asymmetric

	t	t'
Bosco	9	9
W-Bosco	10	9
	11	8
	12	6
	13	5
	14	3
	15	2
	16	0

	t	t'
Bosco-0	12	12
W-Bosco-0	13	11
	14	10
	15	9
	16	8
	17	7
	18	6
	19	5
	20	4
	21	3
	22	2
	23	1
	24	0

With W-Bosco-0, it is clear that we do not need to weaken fault tolerance assumptions as much. Note that with 50 nodes, any consensus protocol can tolerate only up to 16 Byzantine nodes.

ARAGON ZK RESEARCH

Summary so far

We have seen that consensus can be sped up at the cost of tolerating less faulty nodes. We have also seen that, with prior knowledge about the most likely consensus value, the loss of fault tolerance can be limited.

	Symmetric	Asymmetric
Binary consensus	$n > 2t + t'$	$n > 2t + t'$
Weakly one-step	$n > 3t + 2t'$	$n > 2t + 2t'$
Strongly one-step	$n > 3t + 4t'$	$n > 2t + 3t'$

Constraints for consensus in the weakly Byzantine setting

There is an implicit assumption here: at every stage of the protocol, each node must act after receiving n-t messages.

Beyond n-t

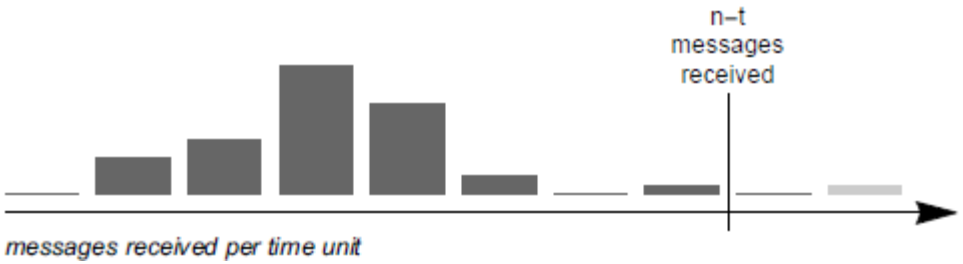
ARAGON
ZK RESEARCH

Beyond n-t

A third approach is to wait for more than the canonical n-t messages. This method can be used in cases when receiving additional messages (1) is likely and (2) will expedite consensus.

Favourable scenario:

Less favourable:



ARAGON
ZK RESEARCH

“Beyond n-t” in the asymmetric case

A one-step decision in the asymmetric case is possible when > t + 2t’ identical messages are received.

If we want to increase fault tolerance for a given n, while still allowing for a one-step decision in some cases, we must choose t and t’ such that:

$$n - t \leq t + 2t' < n$$

For a given n, there are multiple pairs (t, t’) that can be chosen.

ARAGON ZK RESEARCH

Example: 50 nodes

With 50 nodes, we know that a generic consensus protocol can tolerate up to t = t' = 16 Byzantine nodes. Assume we want a protocol that tolerates 16 Crash-prone nodes. Let's see how many Byzantine nodes we can tolerate under different assumptions.

	Waits for:	t	t'
Underlying-consensus	34 msg	16	16
Weakly 1-step, symmetric	34 msg	16	0
Weakly 1-step, asymmetric	34 msg	16	8
Weakly 1-step, asymmetric, wait +7	41 msg	16	12

Conclusion: by waiting for 41 messages instead of 34, one-step consensus can be achieved in favourable conditions, while still tolerating a significant number of crash-prone and Byzantine nodes.

ARAGON
ZK RESEARCH

Example: 13 nodes

With 13 nodes, we know that a generic consensus protocol can tolerate up to t = t' = 4 Byzantine nodes. Assuming we want a protocol that tolerates 4 Crash-prone nodes, we get:

	Waits for:	t	t'
Underlying-consensus	9 msg	4	4
Weakly 1-step, symmetric	9 msg	4	0
Weakly 1-step, asymmetric	9 msg	4	2
Weakly 1-step, asymmetric, wait +2	11 msg	4	3

alex.kampa@aragon.org Telegram @alexkampa WhatsApp +352 691 46 85 81