One-Step Consensus in Weakly Byzantine Environments *

Aleksander Kampa
ak@sikoba.com

Sikoba Research

March 2019
(revised and expanded version)

Abstract

To improve the performance of asynchronous Byzantine consensus, protocols requiring only a single step under favourable conditions have been proposed. We introduce the concept of a *Weakly Byzantine Environment*, in which only a subset of faulty nodes can be Byzantine. We show how conditions for achieving one-step consensus can be relaxed in such environments and propose W-Bosco, a slightly modified version of the Bosco one-step algorithm. W-Bosco introduces an element of choice: if less Byzantine failures can be tolerated, crash resilience can be increased.

1 Introduction

The consensus problem has been called "the most important problem in distributed computing" [CGR11]. Its most generic setting is a fully asynchronous systems with Byzantine faults. Yet it has famously been shown that no deterministic process can guarantee consensus in the presence of a single crash-prone, i.e. not even Byzantine, node [FLP85]. Solutions can still be found by relaxing asynchrony assumptions, injecting randomness, adding "oracles" and/or accepting eventual (i.e. probabilistic) instead of deterministic termination. These solutions often carry a significant processing overhead.

Because in reality failures are rare, significant work has been done to speed up consensus by reducing the number of communication steps in favourable circumstances. The concept of two-step consensus in an asynchronous crash-prone environment appears to have been introduced by Schiper in [Sch97], improving upon the Chandra-Toueg algorithm [CT96] which requires three steps at a minimum. Other two-step algorithms followed, e.g. [KR03]. The concept of one-step consensus was introduced by Brasileiro et al. [BGMR01], again in a crash failure model.

One-step consensus in an asynchronous Byzantine environment was achieved by the Bosco algorithm proposed by Yee Jiun Song and Robbert van Renesse [SvR08]. This was the main inspiration for this paper. Other efforts in that direction have followed, e.g. [BIW10].

In this paper, we introduce the concept of *Weakly Byzantine Environments*, in which only a subset of faulty nodes can be Byzantine, the rest being crash-prone. This could be a realistic assumption in some environments, for example in the context of a consortium blockchain where node operators know each other.

*Research supported by Fantom Foundation

We show how conditions for achieving one-step consensus can be relaxed when only a subset of faulty nodes is Byzantine and introduce W-Bosco, a modified Bosco for weakly Byzantine environments.

One of the contributions of this paper is also its focus on simple and constructive proofs, and its avoidance of proofs by contradiction.

2 Definitions

Node A computer that is part of a network with a common purpose. The computers or nodes forming the network exchange messages with each other in order to reach common decisions.

Protocol, or Algorithm A program that all computers in the network are supposed to apply.

Honest, or Correct Node A computer that is part of the network and applies the protocol as intended.

Faulty Node A computer that is part of the network and may sometimes not apply the protocol as intended.

Crash-prone Node A computer that is usually honest, but may crash and therefore fail to communicate with the other nodes.

Byzantine Node A computer that is not not honest and can send arbitrary messages to honest nodes.

Adversary An intelligent process that has some influence on how faulty nodes behave.

Strong Adversary An intelligent process that is able to coordinate the actions of all faulty nodes and influence the delivery of messages in the network.

Reliable Asynchronous Network A network in which all messages sent eventually get delivered, but with arbitrary delays.

Byzantine Environment An environment in which all faulty nodes are potentially Byzantine.

Weakly Byzantine Environment An environment in which only a subset of faulty nodes is Byzantine, the others being crash-prone.

Consensus Protocol A protocol allowing all correct nodes in a network to decide a single value despite possible faults, and satisfying the following conditions:

- *Agreement*: all correct nodes decide the same value;
- *Validity*: a correct node can only decide a value proposed by another correct node;
- *Termination*: all correct nodes eventually (with probability 1) decide a value.

The agreement and validity properties imply:

- *Unanimity*: if all correct nodes propose the same value v, then this will be the decision value;

Binary Consensus Protocol A protocol in which the input values are 0 or 1. The decision values must therefore also be 0 or 1.

Symmetrical Consensus Protocol A protocol that treats all input values in a similar way, without giving preference to one or the other.

One-step protocol A protocol that allows a decision in one communication step under certain conditions.

Strongly one-step protocol A protocol that allows a decision in one communication step if all correct nodes have the same initial value.

Weakly one-step protocol A protocol that allows a decision in one communication step if all correct nodes have the same initial value and there are no faulty nodes in the system.

One-step decision threshold In the context of a symmetric one-step protocol, the number of votes m out of the first $n - t$ votes received for a value v at which an honest node decides v.

0-node (h_0) An honest node with initial value 0. It will always broadcast the value 0.

1-node (h_1) An honest node with initial value 1. It will always broadcast the value 1.

c_0-node (c_0) A crash-prone node with initial value 0. At any step of the consensus process, it will send the value 0 to some subset of the nodes, following the adversary's instructions.

c_1-node (c_1) A crash-prone node with initial value 1. At any step of the consensus process, it will send the value 1 to some subset of the nodes, following the adversary's instructions.

b-node (b) A Byzantine node that will send different values to every node, following the adversary's instructions.

Configuration An initial configuration is defined in a unique way by the number of: 0-nodes, c0-nodes, Byzantine nodes, c_1-nodes, and 1-nodes. It can be denoted as the set:

$$C = \{h_0,\ c_0,\ b,\ c_1,\ h_1\} \in \mathbb{N}^5 \tag{1}$$

in which:

$$b \leq t' \qquad c_0 + b + c_1 \leq t \qquad h_0 + c_0 + b + c_1 + h_1 = n \tag{2}$$

This implies:

$$n - t \leq h_0 + h_1 \tag{3}$$

0-valent Configuration A configuration in which every correct node will eventually decide 0.

1-valent Configuration A configuration in which every correct node will eventually decide 1.

View For a given Configuration and a given node, it is the set of votes $\{w_0, w_1\}$ for values $\{0, 1\}$ corresponding to the first $n - t$ votes received by that node. Because $w_0 + w_1 = n - t$, any one of w_0 or w_1 suffices to characterise a view.

View Set For a given Configuration, it is the set of all possible Views of any node.

3 Model

We are in the context of an asynchronous distributed system that seeks to reach binary consensus and is subject to both crash and Byzantine failures.

The system consists of n nodes $P = \{p_1, p_2, \dots p_n\}$, each one having an initial value $v_i \in \{0, 1\}$. At most t nodes can be faulty, the remaining nodes are said to be correct.

Among the faulty nodes, at most t' can be Byzantine. The set of $t - t' \geq 0$ faulty nodes that are not Byzantine are crash-prone.

All correct nodes follow a symmetrical binary one-step agreement protocol $\mathcal{A}$ with decision threshold m. As in papers such as [SvR08] or [BIW10] we assume the existence of an **Underlying-Consensus** primitive, which is a binary consensus protocol used as a fallback when a one-step decision is not possible.

The nodes communicate over a reliable asynchronous network.

The adversary has control over the faulty nodes, can see the content of messages and can decide the order in which messages are delivered. However, the adversary cannot impersonate nodes.

4 Preliminary Lemmas

Although several of the following preliminary lemmas are trivially correct, they are included here for completeness sake, as this paper is also intended to be used for didactic purposes.

Lemma 1 At any stage of a consensus protocol, a node must make a decision after receiving $n - t$ votes.

Proof. Given that t nodes might fail, a node may never receive more than $n - t$ votes.

Lemma 2 In a binary one-step consensus protocol $\mathcal{A}$, consider a 0-valent configuration C_0 with View Set V_0 and a 1-valent configuration C_1 with View Set V_1. Then $V_0 \cap V_1 = \varnothing$.

Proof. For each node, $\mathcal{A}$ must decide based on the individual view of that node, and it cannot decide both 0 and 1.

Lemma 3 In the context of a symmetric binary one-step consensus protocol $\mathcal{A}$, assume that one correct node reaches the decision threshold for a value v. Then every node, after receiving $n - t$ votes, will receive a strict majority of votes for v.

Proof Any number of votes for 1 in the range $\big(m \dots n - t\big)$ indicates that we are in a 1-valent configuration. But because nodes tally votes after having received only $n - t$ messages, other nodes may be below the threshold. Let m' be the smallest number of votes for 1 observed in all View Sets of all 1-valent configurations. The range of possible votes for 1 in these configurations is thus $\big(m' \dots n - t\big)$

By symmetry, $\big(m' \dots n - t\big)$ will also be the range of possible votes for 0 observed in all 0-valent configurations. This corresponds to the range $\big(0 \dots n - t - m'\big)$ of votes for 1.

By Lemma 2, the ranges $\big(0 \dots n - t - m'\big)$ and $\big(m' \dots n - t\big)$ do not intersect, and therefore $n - t - m' < m'$, which implies $m' > \frac{n-t}{2}$.

Lemma 4 In the context of a symmetric binary consensus protocol $\mathcal{A}$, given x, y such that $0 \leq x \leq y \leq n - t$ and $y - x \leq t + t'$, there is a configuration in which the range of votes for a given value v over the corresponding view set contains the range $(x \dots y)$.

In other words, when the difference of votes between two views is less or equal to $t + t'$, the algorithm $\mathcal{A}$ is unable to decide if these views correspond to different configurations or not.

Proof: Because the consensus protocol is symmetric, it suffices to show this for w_0.

If $y \leq t'$ we can use the following configuration:

$$\{h_0,\ c_0,\ b,\ c_1,\ h_1\} = \{0,\ 0,\ t',\ 0,\ n - t'\} \tag{4}$$

The honest nodes always vote 1, but the adversary can make as many of the t' Byzantine nodes as necessary vote 0, so the range of w_0 is obviously $(0 \ldots t') \supset (x \ldots y)$.

If $y > t'$ we can use the following configuration:

$$\{h_0,\ c_0,\ b,\ c_1,\ h_1\} = \{y - t',\ 0,\ t',\ 0,\ n - y\} \tag{5}$$

Note that we do not need any crash-prone nodes. The adversary will, however, take advantage of the fact that correct nodes do not know this and must assume that crash-prone nodes might not vote.

To maximise the number of 0-votes, the adversary can make Byzantine nodes vote 0 and delays the delivery of votes by 1-nodes. We can thus achieve $w_0 = y$ in this case:

View max-0

- $y - t'$ honest nodes voting 0)
- t' Byzantine nodes voting 0) votes for 0 : y
- $n - y - t$ honest nodes voting 1) votes for 1 : $n - y - t$
- (delayed: t honest nodes voting 1)

To minimise the number of 0-votes, the adversary makes Byzantine nodes vote 1 and delays 1-nodes. If $y \geq t + t'$, the lowest number of 0-votes is $y - t - t'$ and is achieved in the following view:

View min-0 (if $y \geq t + t$)

- $y - t - t'$ honest nodes voting 0) votes for 0 : $y - t - t'$
- t' Byzantine nodes voting 1)
- $n - y$ honest nodes voting 1) votes for 1 : $n - y + t'$
- (delayed: t honest nodes voting 0)

It is also easily seen that any of the intermediary number of votes can be achieved, so that we get the range of votes $(y - t - t' \ldots y) \supset (x \ldots y)$.

Finally, in the cases where $y < t + t'$, it is also easy to see that we can achieve a view with 0 votes for 0, in which case we have $(0 \ldots y) \supset (x \ldots y)$.

Lemma 5 Consider a set S of n elements, and two subsets S_1 and S_2 with m_1 and m_2 elements such that $m_1 + m_2 > n$. The number of elements in the set $S_1 \cap S_2$ is then at least $m_1 + m_2 - n$.

Proof. Omitted.

5 Minimal Decision Threshold

Theorem 1. The decision threshold m in a symmetrical one-step binary agreement protocol must satisfy $m > \frac{n+t+2t'}{2}$

Proof By Lemma 4, there is a configuration in which the threshold m for value v is reached by one node, but another node receives only $min(0, m-t-t')$ votes, which by the way shows that $m-t-t' > 0$. By Lemma 3, we must have $m-t-t' > \frac{n-t}{2}$. Hence:

$$m > \frac{n+t+2t'}{2} \tag{6}$$

This can be compared with the $\frac{n+3t}{2}$ threshold value in [SvR08] Algorithm 1.

Here is another approach to this. Suppose that node p has received m votes for v. Consider another node q receiving $n-t$ votes. By Lemma 5, the overlap between m and $(n-t)$ is at least $m+(n-t)-n = m-t$. Among these $m-t$ nodes, we could have at most t' Byzantine nodes, so the honest overlap is $m-t-t'$. Or more precisely: $min(0, m-t-t')$. After that we again apply Lemma 3

6 Lower Bound Results

In this section, we restate the lower bound results obtained for standard Byzantine environments by [SvR08].

Lemma 6. A strongly one-step protocol must allow a node to decide v after receiving $n-t-t'$ votes for v

Proof. A node can only wait for $n-t$ messages without blocking. Of these, up to t' can be Byzantine.

Theorem 2. A strongly one-step protocol requires at least $3t+4t'+1$ nodes

Proof. By Lemma 6, a node decides upon receiving $n-t-t'$ votes for a value v. Without loss of generality, we assume that value to be 0. Consider the following initial configuration, which contains the lowest possible number of honest nodes having value 0 and still able to achieve a view with $n-t-t'$ identical votes for 0:

$$\{h_0,\ c_0,\ b,\ c_1,\ h_1\} = \{n-t-2t',\ 0,\ t',\ 0,\ t+t'\} \tag{7}$$

In the following view, the adversary makes all the Byzantine nodes vote 0 and delays t messages from honest 1-nodes. This results in $n-t-t'$ votes for 0:

View max-0

- $n-t-2t'$ honest nodes voting 0)
- t' Byzantine nodes voting 0) votes for 0 : $n-t-t'$
- t' honest nodes voting 1) votes for 1 : t'
- (delayed: t honest nodes voting 1)

In a different view, the adversary makes all the Byzantine nodes vote 1 and delays t messages from honest 0-nodes. This results in $n-t-t'$ votes for 0:

View min-0

- $n - 2t - 2t'$ honest nodes voting 0)
- t' Byzantine nodes voting 1)
- $t + t'$ Byzantine nodes voting 1)
- (delayed: t honest nodes voting 0)

votes for 0 : $n - 2t - 2t'$

votes for 1 : $t + 2t'$

Because of Lemma 3, we must have $n - 2t - 2t' > \frac{(n-t)}{2}$, which implies $2n - 4t - 4t' > n - t$ and therefore $n > 3t + 4t'$.

This can be compared with the $n > 7t$ result in [SvR08] Theorem 1.

Theorem 3. A weakly one-step protocol requires at least $3t + 2t' + 1$ nodes

Proof. In a weakly one-step protocol, a node decides upon receiving $n - t$ identical messages (cf [SvR08] Lemma 2). Without loss of generality, we assume the value received to be 0. Consider the following initial configuration, which contains the lowest possible number of honest nodes having value 0 and still able to achieve a view with $n - t$ identical votes of 0:

$$\{h_0,\ c_0,\ b,\ c_1,\ h_1\} = \{n - t - t',\ 0,\ t',\ 0,\ t\} \tag{8}$$

In a first view, the adversary makes all the Byzantine nodes vote 0 and delays all t messages from the honest 1-nodes. This results in $n - t - t'$ votes for 0:

View max-0

- $n - t - t'$ honest nodes voting 0)
- t' Byzantine nodes voting 0)
- (delayed: t honest nodes voting 1)

votes for 0 : $n - t$

In a second view, the adversary makes all the Byzantine nodes vote 1 and delays t messages from the honest 0-nodes. This results in $n - 2t - t'$ votes for 0:

View min-0

- $n - 2t - t'$ honest nodes voting 0)
- t' Byzantine nodes voting 1)
- t honest nodes voting 1)
- (delayed: t honest nodes voting 0)

votes for 0 : $n - 2t - t'$

votes for 1 : $t + t'$

Because of Lemma 3, we must have $n - 2t - t' > \frac{(n-t)}{2}$, which implies $2n - 4t - 2t' > n - t$ and therefore $n > 3t + 2t'$.

This can be compared with the $n > 5t$ result in [SvR08] Theorem 2.

7 W-Bosco for Weakly Byzantine Environments

7.1 Introducing W-Bosco

The original Bosco algorithm from [SvR08] is shown in Table 1 below.

The main idea is that if the condition in line 3 is true, then it is possible to decide for value v immediately because all other honest nodes will either also decide in line 3, or submit value v to **Underlying-Consensus** (**U-C**) in line 7. By the way, there is an implicit assumption that

deciding for v in line 3 also means that the node in question will signal this as its decision value to **U-C**. As a result, all honest nodes will propose the same value v to **U-C**, which will therefore necessarily decide v because of the *Unanimity* feature.

Table 1: **BOSCO**

	Input: v_p
1	Broadcast (**VOTE**, v_p) to all nodes
2	Wait until $n - t$ **VOTE** messages have been received
3	**if** $> \frac{n+3t}{2}$ **VOTE** messages contain the same value v **then**
4	DECIDE(v)
5	**if** $> \frac{n-t}{2}$ **VOTE** messages contain the same value v,
6	and there is only one such value v **then**
7	$v_p \leftarrow v$
8	**Underlying-Consensus**(v_p)

To obtain W-Bosco, We modify the Bosco algorithm by changing the condition in line 3, and removing line 6, with the following result:

Table 2: **W-BOSCO** for Weakly Byzantine Environments

	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)

We can now prove under which conditions W-Bosco can be strongly or weakly one-step.

Lemma 7. If two correct nodes decide on a value v in line 4, they decide on the same value.

Proof. By Lemma 5, the overlap of two sets of $\frac{(n+t+2t')}{2}$ votes is at least $t + 2t'$. As only t' nodes can be Byzantine, the honest overlap is $t + t'$. If $t + t' = 0$, there cannot be any faulty nodes and the conclusion is trivial. If $t + t' > 0$, the two sets of votes share at least some votes from honest nodes and must therefore be for the same value.

Lemma 8. If a correct node p_i decides a value v in line 4, then another honest node p_j either also decides in line 4 or sets its local estimate to v in line 6.

Proof. By Lemma 5, the overlap between $m > \frac{(n+t+2t')}{2}$ and $n - t$ is $> \frac{(n-t)}{2} + t'$. As only t' nodes can be Byzantine, the honest overlap is $> \frac{(n-t)}{2}$. Node p_j has therefore received strictly more than half of its votes for value v.

Theorem 4. W-Bosco is Strongly One-Step if $n > 3t + 4t'$.

Proof. If all honest nodes have the same initial value v, any node will receive at least $n - t - t'$ votes for this value. This is above the decision threshold value $\frac{(n+t+2t')}{2}$ because:

$$n - t - t' - \frac{(n + t + 2t')}{2} = \frac{(n - 3t - 4t')}{2} > 0 \tag{9}$$

Theorem 5. W-Bosco is Weakly One-Step if $n > 3t + 2t'$.

Proof. If all honest nodes have the same initial value v, and there are no faulty nodes, any node will receive $n - t$ votes for this value. This is above the decision threshold value $\frac{(n+t+2t')}{2}$ because:

$$n - t - \frac{(n + t + 2t')}{2} = \frac{(n - 3t - 2t')}{2} > 0 \tag{10}$$

7.2 Bosco vs W-Bosco: strongly one-step parameters

Let's take a concrete example: a network of 50 nodes. To be strongly one-step, Bosco requires $n > 7t$ and therefore tolerates 7 Byzantine nodes.

The equivalent formula for W-Bosco is $n > 3t + 4t'$, so W-Bosco also tolerates 7 Byzantine nodes when $t = t'$. But other configurations are possible, for example $t = 11$ and $t' = 4$, as $3 * 11 + 4 * 4 = 49 < 50$. If we are prepared to tolerate less Byzantine nodes (4 instead of 7), then we can achieve strong one-step consensus even if a larger number of nodes are crash-prone (11 instead of 7).

All the possible strongly one-step configurations of (t, t') when $n = 50$ are shown in Table 3.

Table 3
Possible parameters for strongly one-step consensus with $n = 50$

	t	t'	$3t+4t'$
Bosco	7	7	49
W-Bosco	7	7	49
	8	6	48
	9	5	47
	11	4	49
	12	3	48
	13	2	47
	15	1	49
	16	0	48

7.3 Bosco vs W-Bosco: weakly one-step parameters

Let's again consider a network of 50 nodes. To be weakly one-step, Bosco requires $n > 5t$ and therefore tolerates 9 Byzantine nodes.

The equivalent formula for W-Bosco is $n > 3t + 2t'$, and with 9 Byzantine nodes we see that it tolerates one additional crash-prone node compared to Bosco, an insignificant improvement. But other configurations are possible, for example $t = 12$ and $t' = 6$. Tolerating less Byzantine nodes (6 compared to 9) again allows us to reach one-step consensus even if a larger number of nodes are crash-prone (12 compared to 10).

All the possible weakly one-step configurations of (t, t') when $n = 50$ are shown in Table 4.

Table 4
Possible parameters 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

8 Conclusion

This paper introduced the concept of *Weakly Byzantine Environments*, in which only a subset of faulty nodes can be Byzantine, the rest being crash-prone.

We showed how conditions for achieving one-step consensus are modified in a Weakly Byzantine Environment and presented W-Bosco, which is a slightly modified version of the Bosco one-step algorithm. W-Bosco does not improve upon Bosco, but it introduces an element of choice: if less Byzantine failures can be tolerated, crash resilience can be increased. This is the paper's main result.

While one-step consensus can only be achieved in environments where the expected proportion of crash-prone and Byzantine nodes is relatively small, we believe that the concept of Weakly Byzantine Environment will be useful in other settings which we plan to explore in the future.

Acknowledgements

I am grateful to Fantom Foundation for their support of Sikoba Research, and to Dr Stefan Beyer for reviewing this paper and providing several useful comments.

Versions

December 2018: first version published on http://research.sikoba.com in December 2018.

March 2019 (current version): expanded version with subsections 7.2 and 7.3 and section 8 (conclusion) added, as well as some other minor changes and corrections.

References

[BGMR01] Francisco Vilar Brasileiro, Fabíola Greve, Achour Mostéfaoui, and Michel Raynal. Consensus in one communication step. In *PaCT*, volume 2127 of *Lecture Notes in Computer Science*, pages 42–50. Springer, 2001.

[BIW10] Nazreen Banu, Taisuke Izumi, and Koichi Wada. Doubly-expedited one-step byzantine consensus. In *DSN*, pages 373–382. IEEE Computer Society, 2010.

[CGR11] Christian Cachin, Rachid Guerraoui, and Luís E. T. Rodrigues. *Introduction to Reliable and Secure Distributed Programming (2. ed.)*. Springer, 2011.

[CT96] Tushar Deepak Chandra and Sam Toueg. Unreliable failure detectors for reliable distributed systems. *J. ACM*, 43(2):225–267, 1996.

[FLP85] Michael J. Fischer, Nancy A. Lynch, and Mike Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, 1985.

[KR03] Idit Keidar and Sergio Rajsbaum. On the cost of fault-tolerant consensus when there are no faults - A tutorial. In *LADC*, volume 2847 of *Lecture Notes in Computer Science*, pages 366–368. Springer, 2003.

[Sch97] André Schiper. Early consensus in an asynchronous system with a weak failure detector. *Distributed Computing*, 10(3):149–157, 1997.

[SvR08] Yee Jiun Song and Robbert van Renesse. Bosco: One-step byzantine asynchronous consensus. In *DISC*, volume 5218 of *Lecture Notes in Computer Science*, pages 438–450. Springer, 2008.