On pseudo-profound bullshit in the Avalanche whitepaper

Ash Ketchum [*]   Misty Williams [*]

November 11, 2019

Abstract

We present an analysis of important claims made in the Avalanche whitepaper *Scalable and Probabilistic Leaderless BFT Consensus through Metastability, Team Rocket et al.* [16]. We find that the consensus protocols described there do not guarantee safety nor liveness and that their conclusions are based on mathematically unsound arguments, pseudo-profound bullshit, and *circulus in probando*. We point out that already a small proportion of malicious nodes can keep the consensus protocol in a metastable state leading to agreement or termination failures. At various places, the arguments in [16] do not follow any scientific standards and can be considered at best inaccurate or misleading.

1 Introduction

Consensus algorithms are a class of algorithms that aim to provide a common decision for all nodes in a networked system and satisfy the following conditions:

1. Agreement: all nodes choose the same value.

2. Termination: all non-faulty nodes eventually decide.

3. Integrity: if the majority of the non-faulty nodes proposes a common value v, then any non-faulty node must choose v.[1]

The paper [16] presents several consensus algorithms that are based on majority dynamics or voter-type models. Let us briefly explain what the main idea of this protocol class is. Essentially every node samples at each discrete time step a random subset of other nodes and queries their states or colors. If the number of answers for one state is above a certain threshold α the node adapts this state. Sampling only a small part (k randomly chosen nodes) of the network

[*] AshMisty@protonmail.com; Nintemoto Labs, 11-1 Hokotate-cho, Manitoba, Minami-ku, Kyoto 601-8501, Japan

[1] Variations of *integrity* may be appropriate and the choice of definition may depend on the applications. The integrity condition is also known as *validity* in the literature.

makes the protocol scalable and the fact that each node queries locally allows certain degrees of asynchronicity in the protocol. These two advantages come however at a price. If a node samples only a small part of the network in the presence of f malicious nodes at each round there will be a positive fraction that queries at least as many malicious than correct nodes.[2] Moreover, these kinds of protocols can usually admit no more than $O(\sqrt{n})$ adversarial nodes; otherwise, the adversary would be able to create an agreement or termination failure, see Section 5. Last but not least, these kinds of protocols in a permission-less network are highly vulnerable against Sybil attacks as an attacker could easily spawn numerous nodes to push f above $n/2$.

Despite the vast literature on this subject no references are given in [16]; we refer to [8, 9, 5, 12] and references therein but note that there are hundreds of papers in this research field. Also note that Team Rocket claims to

> *[...] introduce a brand new family of consensus protocols, based on randomized sampling and metastable decision. [16, page 2].*

The overall whitepaper quality in the cryptocurrency environment is quite low. We are planning to continue our work, exposing bad academic research in this area. What made us start with Avalanche is the particularly enormous gap between their level of bragging, e.g. see Section 7, and the actual quality of their research.[3]

2 Notations and Definitions

The number of nodes is denoted by n and f is the number of adversarial nodes. The $n_c := n - f$ non-adversarial nodes are called correct nodes. We speak of ε-safety if agreement among the correct nodes is achieved with probability at least $1 - \varepsilon$. We speak of ε-liveness if the protocol terminates in finite time with probability $1 - \varepsilon$.[4] These notations correspond to probabilistic versions of agreement and termination.

In Slush (the "foundation" of the Snow family), a node starts out initially in an uncolored state. Upon receiving a transaction from a client, an uncolored node updates its own color to the one carried in the transaction and initiates a first query. To perform a query, a node picks a small, constant sized, k, sample of the network uniformly at random, and sends a query message. Upon receiving a query, an uncolored node adopts the color in the query, responds with that color, and initiates its own query, whereas a colored node simply responds with its current color. In other words, a node either obtains its color externally or from the first node that queries it. Let us define the initial color of a state as its first color and define γ as the initial proportion of red-colored correct nodes. Note here that γ is not a parameter of the protocol.

[2] For instance, if the number of total nodes $n = 1000$, $f = 333$, and $k = 10$, the probability of querying (with replacement) more than $k/2 = 5$ malicious nodes is $P(X \geq 5) \approx 21\%$ where X is a binomial distribution $Binom(10, 1/3)$. In other words, an adversary can control (together with its own nodes) about 47% of all nodes.

[3] If you want to support us in this endeavor you can send BTC to the following address: 333cKk3BxfUVg9NXrCi39NpQy81SRLtzbk

[4] [16] only consider upper and lower bounds for liveness.

Once the querying node collects k responses, it checks if a fraction $\geq \alpha$ are for the same color, where $\alpha > \lfloor k/2 \rfloor$ is a protocol parameter. If the α threshold is met and the sampled color differs from the node's own color, the node flips to that color. It then goes back to the query step, and initiates a subsequent round of query, for a total of m rounds. Finally, the node decides the color it ended up with at time m.

Snowflake augments Slush with a single counter that captures the strength of a node's conviction in its current color. This per-node counter stores how many consecutive samples of the network by that node have all yielded the same color. A node accepts the current color when its counter exceeds β, another security parameter. More precisely, Snowflake incorporates the following modification:

1. Each node maintains a counter *cnt*;

2. Upon every color change, the node resets *cnt* to 0;

3. Upon every successful query that yields $\geq \alpha$ responses for the same color as the node, the node increments *cnt*;

4. A node resets *cnt* to zero if no color appears more than α times in the responses.[5]

Snowball augments Snowflake with *confidence counters* that capture the number of queries that have yielded a threshold result for their corresponding color. A node decides if it gets β consecutive chits for a color. However, it only changes preference based on the total accrued confidence. The differences between Snowflake and Snowball are as follows:

1. Upon every successful query, the node increments its confidence counter for that color.

2. A node switches colors when the confidence in its current color becomes lower than the confidence value of the new color.

The above description of the Snow protocols is essentially copied word by word from [16]. We invite the reader to check the pseudo-codes given in [16] for a better understanding.

3 Non-results of [16]

In the introduction Team Rocket writes:

> *Analysis shows that this metastable mechanism is powerful: it can move a large network to an irreversible state quickly, where the irreversibility implies that a sufficiently large portion of the network has accepted a proposal and a conflicting proposal will not be accepted with any higher than negligible (ε) probability. [16, page1]*

[5] This last point is in the pseudo-code in [16, Figure 5] and not in the corresponding plain text. This may be explained by the fact that a "typo" of [15] was corrected in the pseudo-code only.

They state on their webpage, [1]:

> *Avalanche consensus is a breakthrough consensus protocol that solves some of the issues inherent in the protocols that come before it. It is regarded as the best-of-both worlds*[6] *because it achieves high performance, it is suitable for permissionless settings, and it is able to scale to thousands if not millions of participants.*

Let us investigate what [16] claims to have proven to support the above claims. We quote their main results [16, page 2].

> *Let the system be parametrized for an ε safety failure under a maximum expected f number of adversarial nodes. Let $O(\log n) < t_{max} < \infty$ be the upper bound of the execution of the protocols. The Snow protocols then provide the following guarantees:*
>
> **P1. Safety.** *When decisions are made by any two correct nodes, they decide on conflicting transactions with negligible probability ($\leq \varepsilon$).*
>
> **P2. Liveness (Upper Bound).** *Snow protocols terminate with a strictly positive probability within t_{max} rounds.*
>
> **P3. Liveness (Lower Bound).** *If $f \leq O(\sqrt{n})$, then the Snow protocols terminate with high probability ($\geq 1 - \varepsilon$) in $O(\log n)$ rounds.*

The claim **P1** is circular since the first sentence of the quote already supposes that the system is ε-safe; if the system is ε-safe then it is ε-safe. The condition $O(\log n) < t_{max} < \infty$ makes absolutely no sense, since the O-notation gives upper and not lower bounds, e.g. see [19]. Now, **P2** claims that the protocols terminate with positive probability within a finite time. This a just the minimal condition of any protocol; if a protocol does never converge in finite time it is of no use. A result meaningful and worth to be published would be that the protocols terminate within $O(\log n)$ rounds with probability at least $1 - \varepsilon$. Such a statement is, however, in general wrong for the Snow protocols.

P3 seems to be common knowledge for these kind of consensus protocols, however it fails for $f \geq \lceil \sqrt{n} \log \log n \rceil$, see Section 5. The interesting results would be to understand liveness and safety with more than $\sqrt{n}$ or even a linear proportion of adversarial nodes. We also note that for practicable purposes not only the asymptotic behavior is of importance but also the value of the constants; a point that is completely ignored in [16].

Interesting to note that the above "results" are stated for the Snow protocols but also holds "true" for Slush, which is even according to the authors of [16] unsafe. This raises the question of why there are no results presented on the Snowflake or Snowball protocols in [16]. The answer is that there are no results on these protocols. In the next section, we see how a classic circular argument is applied to convince or fool the reader (and the authors themselves?) on the good performances of these protocols.

[6] classical and Nakamoto consensus

4 Circular reasoning

Circular reasoning is a logical fallacy in which the reasoner begins with what they are trying to end with. Circularity can be difficult to detect if it involves a longer chain of propositions or arguments and concepts not understood by the reasoners. We have already detected circular reasoning in the claim **P1**. This fallacy is extended in the safety study of the protocols. A correct heuristic[7] of the Snow protocols is the following. For any giving set of parameters and number of malicious nodes $f < 1/3n$ if the initial proportion γ of red nodes is outside an interval $[\frac{1}{2} - \delta, \frac{1}{2} + \delta]$ the probability that an adversary can reach agreement or termination failure is very low. The reasoning in [16] follows the argument that the initial proportion γ is a parameter that can be chosen outside this interval $[\frac{1}{2} - \delta, \frac{1}{2} + \delta]$ and they conclude safety and liveness of the protocol. However, the initial proportion γ can not (!) be chosen but is given. The authors of [16] miss completely the point that a protocol has to guarantee safety and liveness for any (!) possible initial proportion γ. In other words, it is assumed that the nodes already share the same colors at the beginning to argue that the protocol reaches consensus among the correct nodes. These observations were also made in [14].

5 Metastability of the Snow protocol family

In physics, metastability is a stable state of a dynamical system other than the system's state of least energy, see [24]. The use of "metastability" in [16] is difficult to understand, see also [11].[8] In the setting of stochastic dynamics, one speaks of a metastable state if the system persists for a long time in this state until it changes to a new equilibrium under the influence of random fluctuations, see [4]. It seems that the authors of [16] mistook the random fluctuations that prevent a system to be stuck in metastable states for metastability.

A reasonable definition of metastable states in our setting could be situations where adversarial nodes can stall the protocol for a long time. In [13] this was achieved for a first version of the Snowball protocol described in [15].[9]

Even after fixing the typo, which by the way had no impact on the theoretical results or "proofs" on the protocol[10], the protocols are likely to be kept in a metastable state by an adversary for a long time. Even worse the "correction of the typo" leads to a very bad convergence speed in other scenarios.

The colors of the nodes in Snowflake and Snowball have the same behavior as the colors in Slush until a first node becomes decided. We assume that $f = \lceil cn^{1/2} \rceil$ for some $c > 0$ and that $\gamma = 0.5$. We also note $n_c = n - f$.

[7] which can also be turned into a rigorous proof

[8] For instance "metastable voting mechanism" and "metastable decision" are meaningless terms here.

[9] `https://twitter.com/AlexSkidanov/status/1132384759437332480`

[10] Changing the protocol from an unsafe to a safe version without changing the theoretical considerations, and claiming that the new version is safe, should make everybody (including the authors) wonder about the validity of the reasoning.

In the first round, half of the adversarial nodes answer red and the other half blue. Let γ_1 be the number of correct red nodes after the first step. Let us see how a node can be red-colored in step 1. There are two possibilities:

1. the node was red at the previous step and it queried less than α blue nodes;

2. the node was blue at the previous step and it queried more than α red nodes.

The number of red nodes after the first step can be seen as the sum of independent Bernoulli distributed random variables. Hence, due to the central limit theorem [20], the distribution of γ_1 can be approximated by a normal distribution $\mathcal{N}(n_c/2, n_c\sigma^2)$. We use the following tail estimates for a gaussian random variable $Z \sim \mathcal{N}(0,1)$: $P(Z > z) \leq \frac{1}{z}e^{-z^2/2}$ for $z > 0$. Now, if $c > \sigma$,

$$\begin{aligned} P\left(\gamma_1 > \frac{n_c}{2} + c\sqrt{n}\right) &\approx P\left(Z > \frac{c\sqrt{n}}{\sigma\sqrt{n_c}}\right) && (1) \\ &\leq \sqrt{\frac{n_c}{n}}\frac{\sigma}{c}e^{-\frac{n}{n_c}\frac{c^2}{2\sigma^2}} \leq e^{-\frac{n}{n_c}\frac{c^2}{2\sigma^2}}. && (2) \end{aligned}$$

If $\gamma_1 \in [\frac{1}{2}n_c - c\sqrt{n}, \frac{1}{2}n_c - c\sqrt{n}]$ then the adversary can adapt the colors of its nodes such that half of the total nodes are red and half of them are blue. We now proceed inductively to define γ_n for $n \geq 2$ and set the corresponding colors of the adversarial nodes. We define the event

$$A_k := \left\{\gamma_k \in \left[\frac{1}{2}n_c - c\sqrt{n}, \frac{1}{2}n_c + c\sqrt{n}\right]\right\}.$$

For $K \in \mathbb{N}$ we obtain

$$P(A_k \; \forall 1 \leq k \leq K) \geq \left(1 - 2e^{-\frac{n}{n_c}\frac{c^2}{2\sigma^2}}\right)^K. \tag{3}$$

If we choose $\frac{c^2}{2\sigma^2} \geq \log K$ we obtain for $K \geq 4$ that

$$P(A_k \; \forall 1 \leq k \leq K) \geq \left(1 - 2e^{-\frac{n}{n_c}\frac{c^2}{2\sigma^2}}\right)^K > \frac{e^{-2}}{2}. \tag{4}$$

Observation 1: *For the Slush protocol there exists a constant $p > 0$ such that for all choices of $\beta, K \in \mathbb{N}$ there exists an adversarial strategy with $f = \lceil\log(K)\sqrt{n}\rceil$ that is able to keep the colors in balance for at least K steps with probability of at least p.*

The above argument may fail for the Snowflake and Snowball protocol since nodes might decide their colors before K steps. Let us therefore give estimates for the first time a node decides its color. We fix $k = 10, \alpha = 8$ and $\beta = 11$[11]. In both protocols a node can only decide

[11] These are parameter choices made in [16].

if it obtains more than α red or α blue answers in β consecutive queries. In the language of probability theory this is known as β runs of a sequence of Bernoulli random variables, see [17]. In fact, in K independent Bernoulli trials with probability of success p, the expected number of runs of length larger than β is $K(1-p)p^{\beta}$. Now, using Markov's inequality, [23], we can bound the probability that at least one node decides before time K by

$$n_c 2K(1-p)p^{\beta}.$$

In our example, $p \approx 0.055$ and hence $(1-p)p^{\beta} \approx 1.2 \cdot 10^{-14}$. For $n = 10.000$ and $K = 1.000.000$ the probability that at least one nodes decides before 1.000.000 steps is less than 0.0003.

Observation 2:[12] *We consider a standard parametrization of the Snowflake or Snowball protocol* $n = 10.000, k = 10, \alpha = 8$ *and* $\beta = 11$. *There exists an adversarial strategy with* $f = \lceil 14\sqrt{n} \rceil = 1400$ *such that no correct node decides its color before one million steps with probability at least* $\frac{e^{-2}}{3}$.

The above probabilities become larger if the number of adversarial nodes is of linear order. Agreement failures can be achieved similarly; the adversary keeps the correct nodes in balance until two correct nodes decide on opposite colors.

6 How to fool yourself or your reader

The academic level of [16] is low. Besides not mentioning the previous relevant work and the fallacies described above, Team Rocket shows a huge lack of understanding. We tried to evaluate the validity of several other statements in [16]. However, in many cases, it is difficult if not impossible to extract non-trivial meaning of the pseudo-profound bullshit [10] and to follow the reasoning of the authors. In most cases, the "results" are as useful as Magikarp [2]. The following is only a random sample since a complete treatment would be beyond the scope of this note.

So let us take a look at one "key" lemma in [16, Appendix]:

> **Lemma 4.** *Slush reaches an absorbing state in finite time almost surely.*

Team Rocket argues that this lemma is a consequence of their Theorem 3 that involves "complicated" formulas and "fancy" notations. However, this statement is one of the first results in every lecture about Markov chains, [22]; any finite (time-homogeneous) Markov chain converges to a stationary distribution, or as in this case to an absorbing state. Moreover, Theorem 3 is only used at this point; so it seems that Team Rocket wanted to fool or impress the audience by fancy notations.

Inequality (5) is wrong, it holds for super-martingales and not for sub-martingales. It is known as the Azuma-Hoeffding inequality, see [18], but given in [16] without any reference.

[12] We do not label these observations as a "Lemma" since in our opinion they really are observations that every reasonable master student following a lecture in probability theory should be able to make.

Note, that this mistake has no impact on the “results” of the paper since inequality (5) is never used anyway! The same holds true for inequalities (3) and (4).

In the justification of inequality (3) Team Rocket seems to have copied [21] almost successfully. However, only writing “$\mathcal{D}(p - \varphi, p)$ is the Kullback-Leiber divergence” without adding, as done in [21], “between Bernoulli distributed random variables [...]” turns it into bullshit.

Section C in [16] is another “highlight”: the authors introduce a lot of notations, e.g. decision function, adversarial strategies, etc., without ever really using them. Moreover, they write “We leave details to the accompanying paper” (bottom of p.17) - but such a paper is not publicly available and, as we have seen in the previous section, will never exist. When we didn’t find the ”accompanying paper”, we tried to look at the original paper [15] which does contain some more ”analysis”; however, since a closer look at it (for example, p.11) reminded us so much of Trubbish [3], we, unfortunately, couldn’t proceed. In any case, we suggest Team Rocket to learn elementary probability theory, see [7] for some good introduction.

Lying with statistics has a long tradition, see [6]. A classical example of misleading graphs uses y-axes with different scales. By carefully adjusting the scales, one can produce surprising trends where none exist or completely distort the facts. While this may seem like an obvious manipulation, one can get away with it because people do not read information. Most people see a graph and immediately conclude from the shape of the lines or bars, exactly as the person who made the graph wants. We present a graph from [16] in Figure 1. This graph is supposed to compare the maximal number of blocks in Bitcoin and the maximal number of rounds required in Snowflake to guarantee a 10^{-20}-safety. One can see that the different scales (in this case the distances between the values $10^1, 10^2$ and 10^3) give a false impression of the superiority of the Snowflake protocol. More surprisingly, the curves of the Snowflake protocol indicate that for $f/n < 0.1$ the number of rounds is negative (!).

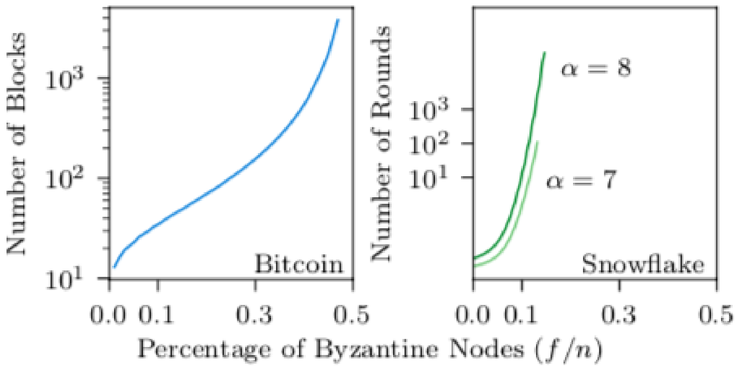

Figure 1: A graphic from [16] supposed to compare the maximal number of blocks in Bitcoin and the maximal number of rounds required by Snowflake to guarantee an 10^{-20}-safety.

7 Closing words

Let us close with a tweet by one of the authors of [16] who is or was Associated Professor at Cornell University giving insights in the usual conversations in Team Rocket:

> *Meanwhile, at the Team Rocket household:*
>
> *"What are you working on dad?"*
>
> *"Nothing, just outdoing that Satoshi fellow on every front, how was your day?"*[13]

Acknowledgement

Ash, a.k.a. Satoshi, would like to thank Pikachu for bringing the above tweet to his attention.

References

[1] Avalabs. https://www.avalabs.org/about, 2019.

[2] Bulpapedia. Magikarp, https://bulbapedia.bulbagarden.net/wiki/magikarp, 2019.

[3] Bulpapedia. Trubbish, https://bulbapedia.bulbagarden.net/wiki/trubbish, 2019.

[4] F. den Hollander. Metastability under stochastic dynamics. *Stochastic Processes and their Applications*, 114(1):1 – 26, 2004.

[5] A. Gogolev, N. Marchenko, L. Marcenaro, and C. Bettstetter. Distributed binary consensus in networks with disturbances. *ACM Trans. Auton. Adapt. Syst.*, 10(3):19:1–19:17, Sept. 2015.

[6] D. Huff. *How to Lie with Statistics.* W. W. Norton and Company, 1993.

[7] Khan Academy. High school statistics, https://www.khanacademy.org/math/probability, 2019.

[8] T. Margush and F. McMorris. Consensus n-trees. *Bulletin of Mathematical Biology*, 43(2):239 – 244, 1981.

[9] E. Mossel, J. Neeman, and O. Tamuz. Majority dynamics and aggregation of information in social networks. *Autonomous Agents and Multi-Agent Systems*, 28(3):408–429, Jun 2013.

[10] G. Pennycook, J. A. Cheyne, N. Barr, D. J. Koehler, and J. A. Fugelsang. On the reception and detection of pseudo-profound bullshit. *Judgment and Decision Making*, 10(6), 2015.

[13] https://twitter.com/el33th4xor/status/1189752696254992393

[11] S. Popov. Iota and avalanche, https://medium.com/@serguei.popov/iota-and-avalanche-35b7cf938664, 2019.

[12] G. Schoenebeck and F. Yu. Consensus of Interacting Particle Systems on Erdös-Rényi Graphs. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2018.

[13] A. Skidanov. snowball, https://github.com/skidanovalex/snowball/, 2019.

[14] A. Skidanov. What avalanche paper is not, https://medium.com/@itsnear/what-avalanche-paper-is-not-c047cf512c16, 2019.

[15] Team Rocket. Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies.
https://ipfs.io/ipfs/QmUy4jh5mGNZvLkjies1RWM4YuvJh5o2FYopNPVYwrRVGV, 2018.

[16] Team Rocket, M. Yin, K. Sekniqi, R. van Renesse, and E. G. Sirer. Scalable and Probabilistic Leaderless BFT Consensus through Metastability. arXiv:1906.08936v1, 2019.

[17] E. W. Weisstein. Run, from MathWorld - A Wolfram Web Resource.
http://mathworld.wolfram.com/run.html, 2019.

[18] Wikipedia contributors. Azuma's inequality, 2019.

[19] Wikipedia contributors. Big O notation, 2019.

[20] Wikipedia contributors. Central limit theorem, 2019.

[21] Wikipedia contributors. Chernoff bound, 2019.

[22] Wikipedia contributors. Markov Chain, 2019.

[23] Wikipedia contributors. Markov's inequality, 2019.

[24] Wikipedia contributors. Metastability, 2019.