# Security threshold for FPCS on star graphs

Rafael Cizeski Nitchai, Serguei Popov, Sebastian Müller, Olivia Saa

# Security threshold for FPCS on star graphs

Rafael Cizeski Nitchai
*Centro de Matemática*
*Faculdade de Ciências*
*Universidade do Porto*
Porto, Portugal
up202008679@up.pt

Serguei Popov
*Centro de Matemática*
*Faculdade de Ciências*
*Universidade do Porto*
Porto, Portugal
serguei.popov@fc.up.pt

Sebastian Müller
*Aix-Marseille Université, CNRS*
Marseille, France
sebastian.muller@univ-amu.fr

Olivia Saa
*IOTA Foundation*
Berlin, Germany
olivia.saa@iota.org

***Abstract*—The Fast Probabilistic Consensus on a Set (FPCS) is a leaderless voting consensus protocol designed for achieving agreement among nodes on a preferred maximal independent set within a graph of conflicts. The protocol's robustness and efficiency have been previously established for complete graphs under the security threshold of $q < \beta < 1/3$, where $q$ represents the proportion of Byzantine nodes. In this paper, we analyze a protocol edge case - a star graph. We show that the security threshold found for complete graphs is not restrictive enough and conjecture a new threshold of $q < \beta < 1/4$. This advance highlights the tradeoff between versatility and reduced security, showing the protocol's adaptability across a broader range of scenarios at the cost of tighter security constraints.**

***Index Terms*—Distributed systems, consensus protocols, byzantine infrastructures.**

## I. Introduction

The Fast Probabilistic Consensus on a Set (FPCS) is a voting consensus protocol designed as the natural evolution of the Fast Probabilistic Consensus (FPC), [1]. Both protocols share several characteristics: they are divided into rounds, are probabilistic in nature (achieving termination, integrity, and agreement with high probability), and their outcomes depend on the existence of a global coin.

While FPC addresses the classical problem of achieving consensus on the value of a bit, FPCS achieves consensus on a Maximal Independent Set (MIS) of a graph of conflicting spendings. This distinction is crucial, as it allows FPCS to manage more complex scenarios, particularly in the context of UTXO-based Distributed Ledger Technologies (DLTs) like Bitcoin, Cardano, Kaspa, and IOTA, where two transactions spending the same output are considered to be in conflict.

Conflicts can arise from malicious behavior but also as the result of faulty node behavior, concurrency and contention in UTXO-based smart contracts, and transaction duplicates when the block times are lower than the network latency.

Despite the differences in the origins of the conflicts, this distinction does not affect our results. We consider a proportion $q$ of nodes to be Byzantine. This encompasses both those actively trying to delay or disrupt consensus (malicious) and those simply defective or unlucky (faulty).

The author Rafael Cizeski Nitchai was partially supported by CMUP, member of LASI, which is financed by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., under the projects with reference UI/BD/150863/2021.

Consider a scenario with three transactions: $u$, $v$, and $w$, where $u$ conflicts with both $v$ and $w$, but $v$ does not conflict with $w$. To update their ledgers, nodes must choose between accepting the set $\{u\}$ or the set $\{v, w\}$ as legitimate. Each round of our protocol generates a set of transactions to be liked by the nodes, which may change significantly in the initial rounds. However, the protocol is designed to stabilize after a few rounds, with nodes consistently liking similar transactions, eventually triggering a stop criterion and rendering the consensus final.

### A. Related Work and Contribution

The total ordering of transactions solves the consensus problem. However, it is shown in [12] that in cases where payments are independent of one another (e.g., UTXO transactions), ordering payments becomes unnecessary. These lower requirements were later observed in multiple papers [11], [13], [14]. Typically, nodes create blocks to approve transactions included in prior blocks, and a quorum of approvals is sufficient to commit a UTXO transaction. While these solutions achieve low communication complexity and latency, they face some practical concerns, as the locking of UTXOs when no quorum of approvals is achieved. Our approach solves the problem of conflicts proactively using a sub-sampling voting procedure.

The protocol and results presented here are relevant to the wider field of majority-dynamics models and particularly interesting to UTXO-based DLTs. By addressing both the classical consensus problem and the complexities of modern transaction conflicts, FPCS offers a different approach to developing consensus mechanisms, which may contribute to more versatile and efficient distributed systems.

## II. Description of the protocol

### A. Notation

Consider a set of $N$ nodes denoted by $\mathcal{N} = \{1, \dots, N\}$ and a set of conflicting transactions $\mathbb{T} = \{u_1, \dots, u_T\}$, which we call the *conflict set*. For our purposes, we assume that a transaction is composed of a unique transaction identifier ("Id", for short), a set of inputs – often referred to as UTXOs (unspent transaction outputs) – and a set of outputs.

We say two transactions $x, y \in \mathbb{T}$ are in conflict if they consume the same UTXO (i.e., if at least one of their inputs is the same) and denote this by $x \leftrightarrow y$. If $x$ and $y$ are not

in conflict, we write $x \nleftrightarrow y$. A transaction $x$ conflicts with a set $B \subset \mathbb{T}$ if it conflicts with every element of $B$ and this is represented by $x \leftrightarrow B$. It is natural to represent the set $\mathbb{T}$ and its conflicts as a graph $G = (\mathbb{T}, E)$, where given $x, y \in \mathbb{T}$ an edge $(x, y) \in E_t$ denotes that $x \leftrightarrow y$.

In this paper, we assume that the structure of the conflict graph can be arbitrary, and depending on the network throughput, the set $\mathbb{T}$ can be very large. A natural way to resolve conflicts is by total ordering of the set of transactions. A popular way to do this is through a *cryptographic hash function* [6] (CHF for short). For the purpose of the paper, we will also assume that the CHF is a pseudo-random function. In particular, our hash function satisfies the property that any random perturbation in the input results in a uniformly distributed independent new output. Let us note that the hash function allows us to define an order on some arbitrary data $x, y$: one can say that $x < y$ if $\text{hash}(x) < \text{hash}(y)$.

Considering discrete time $t = 0, 1, 2 \ldots$ (we refer to it as the round $t$), we define by $A_t^{(n)}$ the set of transactions known by the node $n$ at time $t$ and call it the *node's vision*. Furthermore, we assume that $A_0^{(n)} \subset A_1^{(n)} \subset \ldots \subset \mathbb{T}$ for any $n \in \mathcal{N}$.

We say a node *likes* a transaction if it prefers it to its conflicts. Moreover, we define *node $n$'s opinion* at round $t$ as the collection $O_t^{(n)} = \left\{\theta_t^{(n,x)};\ x \in A_t^{(n)}\right\}$, where $\theta_t^{(n,x)}$ assumes the values:

$$\theta_t^{(n,x)} = \begin{cases} 1, & \text{if node } n \text{ likes transaction } x \text{ at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

For a set $W \subset \mathbb{T}$, we say $\theta_t^{(n,W)} = 1$ if $\theta_t^{(n,w)} = 1$ for all $w \in W$.

We also assume there exists a public sequence of random numbers $X_t \sim U[\beta, 1-\beta]$, which is either provided by a trusted source or generated by the nodes themselves using some decentralized random number generating protocol. This approach is referred to as a *global coin* in many works on Byzantine consensus, for example, in [7]–[10]. We assume all random numbers and messages between the nodes are delivered on time in every round.

Our objective is to formulate a protocol that facilitates consensus among the nodes $\mathcal{N}$ regarding a Maximal Independent Set (MIS) within $\mathbb{T}$. A designated proportion $q$ of the nodes, referred to as *malicious*, may opt not to adhere to our protocol, thereby choosing to impede or disrupt the consensus process. For a constant $c \in [0, 1]$, we say the protocol is *resistant* up to a threshold $c$ if, for any $q < c$, consensus can be achieved with high probability.

Without loss of generality, we assume that the first $(1-q)N$ nodes are *honest* (i.e., not malicious) and define the proportion of likes among honest nodes of a transaction $u \in \mathbb{T}$ as

$$p_t^{(u)} := \frac{1}{(1-q)N} \sum_{j=1}^{(1-q)N} \theta_t^{(j,u)}. \tag{1}$$

For a set $U \in \mathbb{T}$, the proportion of likes $p_t^{(U)}$ is defined as the proportion of honest nodes that like every $u \in U$.

We define the *Interval of Control* of the malicious nodes over a transaction $v$ at round $t$ as

$$\mathcal{I}_{q,t}^{(v)} := [(1-q)p_t^{(v)}, (1-q)p_t^{(v)} + q].$$

The lower/upper boundary of this interval is precisely the overall proportion of likes (i.e., considering both honest and malicious opinions) that the transaction has when all malicious nodes dislike/like it. Malicious nodes can, thus, control the overall proportion of likes of a transaction within this interval.

Our results hinge on the observation that once a significant majority of honest nodes align on a specific transaction or set of transactions, it becomes difficult for malicious nodes to reverse this opinion. To precisely delineate the threshold for a significant majority, we introduce the abbreviation

$$\mu := \frac{\beta - q}{2(1-q)}.$$

Fig. 1 illustrates the concept of Intervals of Control and the relations between $\beta$ and $q$. In particular, we will assume that $q < \beta$, or in other words, that $\mu$ is positive.

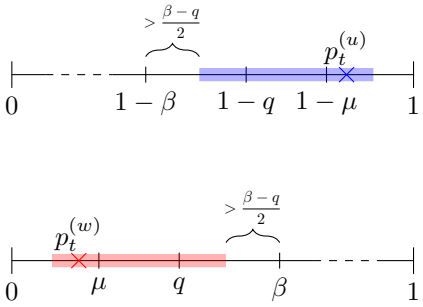


Fig. 1. The intervals of control $\mathcal{I}_{q,t}^{(u)}$ for a transaction $u$ which has a proportion of likes among honest nodes $p_t^{(u)} > 1-\mu$ (in blue), and $\mathcal{I}_{q,t}^{(w)}$ for a transaction $w$ such that $p_t^{(v)} < \mu$ (in red). Notice that these intervals are separated from the support $[\beta, 1-\beta]$ of $X_t$.

### B. Protocol

For $t = 0$, the initial visions, $A_0^{(n)}$, and opinions, $O_0^{(n)}$, can be arbitrary, as long as the set of liked transactions for each node forms a maximal independent set of the conflict graph induced by $A_0^{(n)}$. Then, the following protocol should be executed by each node iteratively once for every round $t \geq 1$ until the stop criterion is met:

1) Query $k$ (uniformly) random nodes[1] about their liked transactions.
2) Store any received transactions it was not aware of.
3) The node, denoted by $n$, stores the collection $\left\{\eta_t^{(n,x)};\ x \in A_t^{(n)}\right\}$, where $\eta_t^{(n,x)}$ corresponds to the number of 1-opinions it received from the queries in round $t$ with respect to the transaction $x$.
4) Receive a random value $X_t \sim U[\beta, 1-\beta]$[2].

[1]Every node can use its own source of randomness.
[2]Every nodes receives the same random value.

5) Define an auxiliary collection of opinions $\left\{\theta'(x);\ x \in A_t^{(n)}\right\}$, that will not be shared and will last only until the end of the round (hence we omit the dependence on $n$ and $t$), using the following rule:

$$\theta'(x) = \begin{cases} 1, & \text{if } \eta_t^{(n,x)}/k > X_t \\ 0, & \text{otherwise.} \end{cases}$$

6) Let $B := \left\{x \in A_t^{(n)};\ \theta'(x) = 1\right\}$. The node must find a way to assign 1 only to the opinions of a maximal independent subset of $A_t^{(n)}$. To do so, it iteratively removes from $B$ the transaction $x \in B$ with the largest $\mathrm{hash}(\mathrm{Id}_x, X_t)$ (this means the hash of $\mathrm{Id}_x$ concatenated with the random number $X_t$) until it obtains an independent set. Note that using the "largest hash" is not crucial, as any deterministic rule leading to unpredictable results is sufficient. Explicitly, it performs the following algorithm: Consider $B' := \mathrm{elim}(B, X_t)$.

**Algorithm 1** $\mathrm{elim}(U, X_t)$

$W = U$
**while** $W$ is not an independent set **do**
  Compute
$$y = \underset{x \in W : \exists z \in W : z \leftrightarrow x}{\operatorname{argmax}} \mathrm{hash}(\mathrm{Id}_x, X_t),$$
  $W = W \setminus \{y\}$
**end while**
**return** $W$

While this set is independent by construction, it may not be maximal. Then, starting with $B'$, the node includes iteratively the non-conflicting transaction with the smallest $\mathrm{hash}(x, X_t)$ until a maximal independent set is obtained. Explicitly, the node executes the following: Let $B'' := \mathrm{compl}(B', A_t^{(n)}, X_t)$. Finally, the node

**Algorithm 2** $\mathrm{compl}(U, V, X_t)$

$W = U$
**while** $W$ is not a maximal independent set **do**
  Compute
$$y = \underset{x \in V \setminus N(W,V)}{\operatorname{argmin}} \mathrm{hash}(x, X_t)$$
  $W = W \cup \{y\}$
**end while**
**return** W

assigns value 1 to the opinion $\theta_{t+1}^{(n,x)}$ of every transaction $x \in B''$ and zero to the others.

If the node's opinion about a transaction does not change for $\ell$ rounds, then it is considered final and will not be further modified in the subsequent rounds.

## III. Results

In [2], the specific scenario where $\mathbb{T}$ represented a complete graph was studied, yielding robust findings regarding consensus within a security threshold $q < \beta < 1/3$.

While complete graphs, commonly known as $n$-spends in classical cryptocurrency literature, represent a typical form of attack, they do not constitute the most sophisticated form. In more intricate situations, malicious nodes extend their influence beyond mere voting, manipulating $\mathbb{T}$ itself—for instance, by introducing new conflicting transactions—to advance their objectives. We explore now an edge case in which the threshold $q < \beta < 1/3$ proves insufficient to guarantee consensus.

### A. Star Graphs

A $S_j$ star graph is a complete bipartite graph constituted by one internal vertex connected to a set of $j$ external vertices, called leaves. Of course, the two only possible maximal independent sets in this graph are the set of leaves and the singleton of the interior vertex.

Assume that $\mathbb{T} = S_j$ for some integer $j \geq 2$ and designate $u$ as the interior vertex. Consider also that $k = N$ or, in other words, that nodes will query every other node every round. Assume malicious nodes will adopt the following strategy: when queried by a node that likes $u$ (resp. $N(u)$), the malicious will reply it also likes $u$ (resp. $N(u)$). Moreover, consider that $p_t^{(u)} = 1/[2(1-q)]$ and define as $\hat{p}_t^{(x)}$ and $\tilde{p}_t^{(x)}$ the overall proportion of likes (i.e. including the opinions of malicious nodes) that a transaction $x \in \mathbb{T}$ has at round $t$ from the perspective of a node that likes $u$ and $N(u)$ respectively.

Now consider the case $1/6 < q < \beta < 1/3$ where malicious nodes control a significant, though not critical, proportion of nodes. Then it can easily be verified (see Fig. 2) that

$$\tilde{p}_t^{(N(u))} = p_t^{(N(u))}(1-q) < \beta;$$
$$\tilde{p}_t^{(u)} = p_t^{(u)}(1-q) + q > 1 - \beta;$$
$$\hat{p}_t^{N(u)} = p_t^{(N(u))}(1-q) + q = p_t^{(u)}(1-q) = \hat{p}_t^{(u)} = 1/2.$$

From the first two relations, we observe that regardless of the outcome of $X_t$, nodes that liked $u$ at the beginning of the round will see no reason to change their mind since $\tilde{p}_t^{(u)} > 1 - \beta > X_t > \beta > \tilde{p}_t^{(N(u))}$.

On the other hand, nodes that originally liked $N(u)$ will encounter a tie $\hat{p}_t^{(u)} = \hat{p}_t^{(N(u))} = 1/2$ and then for every $x \in \mathbb{T}$ either assign $\theta'(x) = 1$ if $X_t < 1/2$, or assign $\theta'(x) = 0$ if $X_t > 1/2$. In both cases, step 7) of our protocol will pick between $\{u\}$ and $N(u)$, the set that contains the transaction with the smallest $\mathrm{hash}(\mathrm{Id}_x, X_t)$. Due to the uniformity property of the hash function, the smallest hash will be in $N(u)$ with probability $j/(j+1)$. This implies that malicious nodes can, with high probability, bypass the random component of the protocol and compel nodes that liked $N(u)$ initially to persist in liking $N(u)$. If this situation persists for $\ell$ rounds, opinions become final, and consensus is broken.

This attack is only possible because the union of the intervals of control $\mathcal{I}_{q,t}^{(u)} \cup \mathcal{I}_{q,t}^{(N(u))}$ covers the whole support of $X_t$. An intuitive way to solve this is to decrease $\beta$ (and consequently enlarge the support of $X_t$), but by doing that, since $q$ must be smaller than $\beta$, we are also getting less resistant.

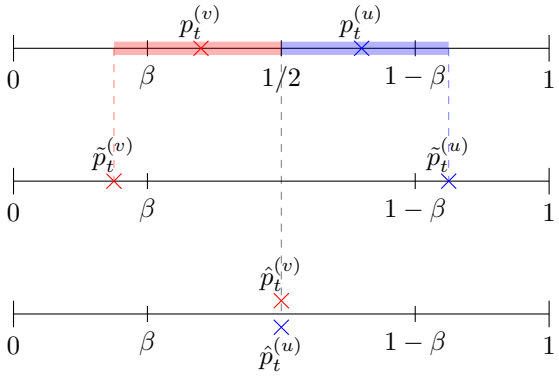


Fig. 2. The interval of control in blue for the interior edge $u$, and in red for a leaf transaction $v \in N(u)$. Notice that the union of the intervals of control covers the whole support of $X_t$.

To find a middle ground, we decrease $\beta$ by a margin just enough to guarantee that the support of $X_t$ is larger than $2q$ (two times the length of an interval of control). In other words, we want to maximize $\beta$ subject to the constraints $2q < 1-2\beta$ and $q < \beta$. The result is the security threshold $q < \beta < 1/4$.

A fundamental property of the system under this security threshold is that, under certain outcomes of $X_t$, nodes are not only likely to approve any transaction $u \in \mathbb{T}$ that has a sufficiently large proportion of likes but also, at the same time, disapprove any transaction in $N(u)$.

To find exactly how large this proportion of likes has to be, notice that if $u \in \mathbb{T}$ has $p_t^{(u)} = (1-\beta-q)/(1-q)$ then the upper boundary of $\mathcal{I}_{q,t}^{(u)}$ is $1-\beta$ (which is equal to the upper boundary of the support of $X_t$); on the other hand, if $p_t^{(u)} = 1/[2(1-q)]$, then the lower boundary of $\mathcal{I}_{q,t}^{(u)}$ is $1/2$. We define $p^*$ as the middle point between these two values:

$$p^* := \frac{1}{2}\left[\frac{1-\beta-q}{1-q} + \frac{1}{2(1-q)}\right] = \frac{1}{2(1-q)} + \frac{1-2\beta-2q}{4(1-q)}.$$

This way if $p_t^{(u)} < p^*$ (resp. $p_t^{(u)} \geq p^*$) there will be a gap of at least size $h := (1/2-\beta-q)/2$ between $\mathcal{I}_{q,t}^{(u)}$ and $1-\beta$ (resp. $1/2$). This last property motivates the following conjecture.

**Conjecture III.1.** *Under the security threshold $q < \beta < 1/4$, the FPCS protocol is capable of achieving consensus on a MIS within an arbitrary conflict graph $\mathbb{T}$ with high probability.*

The basis for this conjecture arises from the following observation: The elim/compl steps of the algorithm, for any graph, select a maximal independent set (MIS) at random. In the case of complete graphs, the preferred transaction is the one with the smallest hash, resulting in a uniformly random selection of the MIS. For star graphs, however, the MIS is not chosen uniformly at random but is influenced by the number of leaves in the star. This is what allows attackers to circumvent the randomness of the protocol when $1/6 \leq q < \beta < 1/3$. Importantly, the dependence of the MIS selection by the elim/compl step on the size of the MIS is not restricted to star graphs, but a general property. Furthermore, it is also a general property that the union of the intervals of control $\mathcal{I}_{q,t}^{(u)} \cup \mathcal{I}_{q,t}^{(N(u))}$ over any transaction $u$ does not fully cover the support of $X_t$ under the security condition $q < \beta < 1/4$.

## IV. Conclusion and Future Work

This paper describes a vector of attack for the FPCS protocol in which consensus is broken considering $q < \beta < 1/3$. We propose an alternative security threshold $q < \beta < 1/4$ for which this attack is mitigated. Furthermore, we conjecture that this new threshold is sufficient to guarantee consensus for an arbitrary conflict graph with high probability.

A critical feature of our protocol is its reliance on a sequence of random numbers $X_t$,. We believe that perfect randomness is not required, and future research will aim to rigorously demonstrate this aspect. Additionally, we aim to relax the synchronicity assumptions of the underlying communication system. The inherent randomness of the subsampling makes the protocol robust to different local perceptions, and this intuition should be made more rigorous in future research.

## References


[1] Serguei Popov, William J. Buchanan, FPC-BI: Fast Probabilistic Consensus within Byzantine Infrastructures, Journal of Parallel and Distributed Computing, Volume 147, 2021, Pages 77-86, ISSN 0743-7315,

[2] Nitchai, R.C., Popov, S., Müller, S. (2023). FPCS: Solving n-Spends on a UTXO-Based DLT. In: Machado, J.M., et al. Blockchain and Applications, 5th International Congress. BLOCKCHAIN 2023. Lecture Notes in Networks and Systems, vol 778. Springer, Cham.

[3] Müller, S., Penzkofer, A., Kuśmierz, B., Camargo, D., Buchanan, W.J. (2021). Fast Probabilistic Consensus with Weighted Votes. In: Arai, K., Kapoor, S., Bhatia, R. (eds) Proceedings of the Future Technologies Conference (FTC) 2020, Volume 2 . FTC 2020. Advances in Intelligent Systems and Computing, vol 1289. Springer, Cham.

[4] Angelo Capossele, Sebastian Müller, Andreas Penzkofer, Robustness and efficiency of voting consensus protocols within byzantine infrastructures, Blockchain: Research and Applications, Volume 2, Issue 1, 2021, 100007, ISSN 2096-7209,

[5] Müller, S., Penzkofer, A., Camargo, D., Saa, O. (2021). On Fairness in Voting Consensus Protocols. In: Arai, K. (eds) Intelligent Computing. Lecture Notes in Networks and Systems, vol 284. Springer, Cham.

[6] Merkle, Ralph. (1989). One Way Hash Functions and DES. 428-446.

[7] Cachin, C., Kursawe, K. Shoup, V. Random Oracles in Constantinople: Practical Asynchronous Byzantine Agreement Using Cryptography. J Cryptology 18, 219–246 (2005).

[8] Canetti, R. and Rabin, T. (1993). Fast asynchronous Byzantine agreement with optimal resilience. In Proceedings of the 25th Annual ACM Symposium on Theory of Computing, pages 42–51.

[9] Marcos K Aguilera and Sam Toueg. The correctness proof of Ben-Or's randomized consensus algorithm. Distributed Computing, 25(5):371–381, 2012.

[10] Friedman, R., Mostefaoui, A., and Raynal, M. (2005). Simple and efficient oracle-based consensus protocols for asynchronous Byzantine systems. IEEE Transactions on Dependable and Secure Computing, 2(1):46–56.

[11] Mathieu Baudet, Alberto Sonnino, Mahimna Kelkar, and George Danezis. Zef: low-latency, scalable, private payments. Proceedings of the 22nd Workshop on Privacy in the Electronic Society, pages 1–16, 2023.

[12] Rachid Guerraoui, Petr Kuznetsov, Matteo Monti, Matej Pavlovič, and Dragos-Adrian Seredinschi. The consensus number of a cryptocurrency. Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, pages 307–316, 2019.

[13] Andrew Lewis-Pye, Oded Naor, and Ehud Shapiro. Flash: An Asynchronous Payment System with Good-Case Linear Communication Complexity. arXiv preprint arXiv:2305.03567, 2023.

[14] Mathieu Baudet, George Danezis, and Alberto Sonnino. Fastpay: High-performance byzantine fault tolerant settlement. Proceedings of the 2nd ACM Conference on Advances in Financial Technologies, pages 163–177, 2020.