



The sum of its parts: Analysis of federated byzantine agreement systems

Martin Florian¹ · Sebastian Henningsen¹ · Charmaine Ndolo¹ · Björn Scheuermann²

Received: 27 November 2020 / Accepted: 8 June 2022 / Published online: 12 July 2022
© The Author(s) 2022

Abstract

Federated Byzantine Agreement Systems (FBASs) are a fascinating new paradigm in the context of consensus protocols. Originally proposed for powering the Stellar payment network, FBASs can instantiate Byzantine quorum systems without requiring out-of-band agreement on a common set of validators; every node is free to decide for itself with whom it requires agreement. Sybil-resistant and yet energy-efficient consensus protocols can therefore be built upon FBASs, and the “decentrality” possible with the FBAS paradigm might be sufficient to reduce the use of environmentally unsustainable proof-of-work protocols. In this paper, we first demonstrate how the robustness of individual FBASs can be determined, by precisely determining their safety and liveness buffers and therefore enabling a comparison with threshold-based quorum systems. Using simulations and example node configuration strategies, we then empirically investigate the hypothesis that while FBASs can be bootstrapped in a bottom-up fashion from individual preferences, strategic considerations should additionally be applied by node operators in order to arrive at FBASs that are robust and amenable to monitoring. Finally, we investigate the reported “open-membership” property of FBASs. We observe that an often small group of nodes is exclusively relevant for determining liveness buffers and prove that membership in this top tier is conditional on the approval by current top tier nodes if maintaining safety is a core requirement.

Keywords Byzantine quorum systems · Asymmetric trust · Byzantine faults · Consensus · Stellar · Blockchain

1 Introduction

We study *Federated Byzantine Agreement Systems* (FBASs), as originally proposed by Mazières [16]. FBASs are conceptually related to *Asymmetric Quorum Systems* [2] and *Personal Byzantine Quorum Systems* [14]. While research on consensus protocols has accelerated in the wake of global blockchain enthusiasm, developments still mostly fall in two extreme categories: *permissionless*, i.e., open-membership, as exemplified by Bitcoin’s notoriously energy-hungry “Nakamoto consensus” [17], and *permissioned*, with a closed group of validators, as assumed both in the classical Byzantine fault tolerance (BFT) literature (e.g., [4]) and many state-of-the-art protocols from the blockchain world (e.g., [22]). The FBAS paradigm and the works it has inspired

suggest a middle way: *Each* node defines its *own rules* about which groups of nodes it will consider as sufficient validators. If the sum of all such configurations fulfills a set of properties, protocols like the *Stellar Consensus Protocol* (SCP) [16] can be defined that leverage the resulting structure for establishing a live and safe consensus system [3,8,9,13,14].

In the original FBAS model [16], which this paper is based on, these properties are foremost *quorum availability despite faulty nodes*, which enables *liveness*, and *quorum intersection despite faulty nodes*, which makes it possible for consensus protocols to prevent forks and thus enables *safety*. In a practical deployment, it is seldom clear which nodes are faulty, and in this way the level of risk w.r.t. to liveness and safety is uncertain. We propose an intuitive and yet precise analysis approach for determining the level of risk, based on enumerating *minimal blocking sets* and *minimal splitting sets*—minimal sets of nodes that, if faulty, can by themselves compromise *liveness* and *safety*. We provide algorithms for determining these sets in arbitrary FBASs and make available

✉ Martin Florian
martin.florian@hu-berlin.de

¹ Humboldt-Universität zu Berlin / Weizenbaum Institute, Berlin, Germany

² Technische Universität Darmstadt, Darmstadt, Germany

an efficient software-based analysis framework¹. To the best of our knowledge, we are the first to propose and implement an analysis methodology for the assessment of the liveness and safety guarantees of FBAS instances that yields precise results as opposed to heuristic estimations. As previously shown in [8], FBASs induce *Byzantine quorum systems* as per Malkhi and Reiter [15]—hence our results might be of interest to more classical formalizations as well. For example, we explicitly distinguish between sets of nodes that can undermine liveness and such sets that can undermine safety, highlighting that in an actual system the threat to liveness and the threat to safety can differ both in structure and in severity.

We apply our analysis approach and tooling in an empirical study that investigates the emergence of FBASs from existing inter-node relationships, as encoded in, e.g., trust graphs. Based on example *configuration policies*, we demonstrate that while FBASs can be bootstrapped in a bottom-up fashion from individual preferences, strategic considerations should additionally be applied by node operators in order to arrive at FBASs that are robust and amenable to monitoring.

Strategic considerations can increase centralization, on top of what is already implied by individual preferences. We observe that centralization manifests as a *top tier* of nodes that is solely relevant when determining liveness buffers. We contribute a proof that if maintaining basic safety guarantees is a minimal strategic requirement of node operators, top tiers are effectively “closed-membership” in the sense that a top tier’s composition can only change with cooperation of current top tier nodes. This casts doubt on the reported “open-membership” property of FBASs—while any node can become part of the FBAS, our results show that only nodes approved by the current top tier can become relevant for consensus.

Following an overview of related work (Sec. 2) and the formal introduction of the FBAS model and its interpretation in practical deployments (Sec. 3), we structure our paper around our main original contributions:

- An analysis framework for reasoning about safety and liveness guarantees in concrete FBASs (Sec. 4).
- Algorithms for efficiently performing the proposed analyses (Sec. 5).
- A simulation-based exploration of possible configuration policies and their effects (Sec. 6).
- Formal proof that membership in an FBAS’ top tier is only “open” if a violation of safety is considered acceptable (Sec. 7).

As appendices, we prove a number of additional corollaries and theorems (Appendix A) and present results from applying our analysis methodology to an interesting toy

network (Appendix B) and the current Stellar network (Appendix C).

2 Related work

Federated Byzantine Agreements Systems were first proposed in [16], together with the *Stellar Consensus Protocol* (SCP), a first protocol for this setting. The viability of SCP has been proven formally [8,9,13] and the protocol is in active use in two large-scale payment networks [13,18]. The FBAS notion has furthermore been generalized and reformulated in different ways, creating bridges to more classical models and enabling the development of additional protocols [2,3,14]. Among other things, as shown by García-Pérez and Gotsman [8], FBASs with “safe” configurations induce Byzantine quorum systems [15]. In this work, we are less interested in the mechanics of specific protocols for the FBAS setting but instead investigate the *conditions* they require for achieving safety, liveness and performance. We investigate how many node failures (and of which nodes) an FBAS can tolerate before the conditions to safety and liveness are compromised, and how individual node configuration policies influence these “buffers”.

Previously, consensus protocols relevant in practice (such as PBFT [4]) have relied on a symmetric threshold model. In a typical instantiation with $3f + 1$ nodes that can tolerate up to f Byzantine node failures, each $2f + 1$ nodes form a (minimal) quorum. This model naturally gives rise to quorum systems that are trivial to analyze, i.e., for which it is trivial to determine under which maximal fail-prone sets [15] consensus is still possible. The possibility for quorum systems that lack symmetry (that is opened up by the FBAS paradigm and related notions) makes the investigation of a more general analysis approach necessary.

A heuristics-based methodology for analyzing FBAS instances was previously proposed in [11], focusing on the identification of central nodes and threats to FBAS liveness. We propose a novel analysis approach that is not heuristics-based and hence yields precise insights, based on a solid theoretic foundation. As in [11], we apply our methodology to snapshots of the live Stellar network (cf. Appendix C).

Bracciali et al. [1] explore fundamental bounds on the decentrality in open quorum systems. One of their central arguments with regards to the FBAS paradigm is that quorum intersection, a crucial requirement to guaranteeing safety in protocols like SCP, is computationally intractable to determine and maintain, necessitating centralization if safety is a requirement. The NP-hardness of determining quorum intersection was previously also proven by Lachowski [12], together, however, with practical algorithms for nevertheless determining safety-critical properties of non-trivial FBASs. We develop new algorithms that incorporate the possibility

¹ https://github.com/wiberlin/fbas_analyzer

that some nodes may fail, enumerating *minimal blocking sets* and *minimal splitting sets*. We evaluate their performance for different FBAS sizes, providing insights into the computational limitations that are relevant in practice. While, based on our analysis approach and its application to specific FBASs, we can confirm that nodes of higher influence (*top tier* nodes according to our choice of words) naturally emerge, we argue that it is not only the existence and size of such a group that determines “centralization” but also the fluidity of that group’s membership (which we explicitly investigate).

An alternative analysis methodology and software framework has recently been presented in [10]. Among other things, the authors provide algorithms for determining the consequences of specific sets of nodes becoming faulty, whereas we propose and implement approaches for identifying *all* minimal sets of nodes that need to become faulty for an FBAS to lose safety and liveness guarantees.

3 Federated byzantine agreement

In the following, we introduce core concepts of the FBAS paradigm that form our basis for reasoning about specific FBAS instances. We use terminology based on [12,13,16] and the Stellar codebase (*stellar-core*).

Our FBAS model is based on the concept of *nodes*. Whereas nodes usually represent individual machines, for the purposes of this paper we typically assume that each node represents a distinct entity or organization. We will illustrate introduced concepts using examples, with nodes represented as integers. For example, $\{0, 1, 2\}$ denotes a set of three distinct nodes. We will occasionally also use established terms in the context of consensus protocols, such as “slot”, “externalize” and “faulty”, without formally introducing them. As an informal and approximate adaptation to the blockchain setting, a *slot* is a block of a given height, to *externalize* a *value* is to decide the contents of a block², and a *faulty* node is one that violates protocol rules in arbitrary ways, e.g., assuming the worst-case scenario, via being under the control of an attacker that also controls all other faulty nodes.

We first introduce the formal foundation of the FBAS paradigm as originally proposed in [16]. Following that, we formally define the *quorum set* configuration format for FBAS nodes that was previously only used in a practical implementation (of the Stellar network software) but whose convenience for defining specific FBAS instances also benefits the theoretical discussion. Based on the introduced foundations, we finally derive the necessary properties an

FBAS must exhibit in order to enable liveness and safety guarantees.

3.1 Quorum slice and FBAS

In an FBAS, each node (respectively its human administrator) individually configures which other nodes’ opinions it should consider when participating in consensus. Configurations can express individual expectations, such as “out of these n nodes, at most f will simultaneously cooperate to attack the system”, and can be used to strategically influence global system parameters. On a conceptual level, the configuration of an FBAS node consists in the definition of *quorum slices*.

Definition 3.1 (*FBAS; adapted from [16]*) A Federated Byzantine Agreement System (FBAS) is a pair (\mathbf{V}, \mathbf{Q}) comprising a set of nodes \mathbf{V} and a quorum function $\mathbf{Q} : \mathbf{V} \rightarrow 2^{2^{\mathbf{V}}}$ specifying *quorum slices* for each node, where a node belongs to all of its own quorum slices—i.e., $\forall v \in \mathbf{V}, \forall q \in \mathbf{Q}(v), v \in q$.

Informally, each quorum slice of a node v describes a set of nodes that, should they all agree to externalize a value in a given slot, is sufficient to also cause v to externalize that value.

Clearly, an FBAS cannot be modeled as a regular graph (with FBAS nodes as graph edges) without losing information. Graph-based analyses as in [11] can therefore result only in heuristic insights. An FBAS *can* be modeled as a directed hypergraph [7]. However, we find the *quorum set* abstraction (presented next) more suitable for subsequent analysis. In Sec. 6, we explore strategies for bootstrapping robust FBASs from graphs.

3.2 Quorum set

While a useful abstraction for formally describing protocols for the FBAS setting, quorum slices are an unwieldy format for describing concrete FBAS instances. In Stellar, the currently most relevant practical deployment of an FBAS, nodes are configured not via quorum slices but via *quorum sets* [13]. Each quorum set defines a set of validator nodes $U \subseteq \mathbf{V}$, a set of inner quorum sets \mathcal{I} and a threshold value t . Intuitively, this representation enables the encoding of notions such as “out of these nodes U , at least t must agree” (*satisfying* the quorum set) or “the sum of agreeing nodes in U and satisfied inner quorum sets in \mathcal{I} must be at least t ”.

Definition 3.2 (*quorum set; adapted from Stellar codebase*) A quorum set is a recursive tuple $(U, \mathcal{I}, t) \in \mathfrak{D}$, $\mathfrak{D} := 2^{\mathbf{V}} \times 2^{\mathfrak{D}} \times \mathbb{Z}^+$. For quorum sets of the form $D = (U, \mathcal{I}, t)$, we recursively define that a set of nodes $q \subseteq \mathbf{V}$ *satisfies* D iff $(|q \cap U| + |\{I \in \mathcal{I} : q \text{ satisfies } I\}|) \geq t$.

² Consensus protocols for the FBAS setting typically provide *immediate finality*, in the sense that once the value for a slot has been externalized, it cannot be reverted or changed.

For example, $(\{0, 1\}, \emptyset, 1)$ encodes that agreement is required from either node 0 or node 1, whereas $(\{0\}, \mathcal{I}, 1)$ with $\mathcal{I} = \{(\{1, 2, 3\}, \emptyset, 2)\}$ encodes that either node 0 or two out of $\{1, 2, 3\}$ must agree. Inner quorum sets (members of \mathcal{I}) are often used for grouping nodes belonging to the same entity (respectively organization), so that the importance of an entity can be decoupled from the number of nodes it controls.

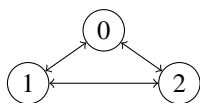
Quorum sets are useful for defining the quorum slices of a node. To ease notation, we define the formalism $\text{qset}(v, D)$ that expresses the set of quorum slices of a node $v \in \mathbf{V}$ based on a quorum set $D \in \mathcal{D}$.

Definition 3.3 (*quorum set \rightarrow quorum slices*) For a node $v \in \mathbf{V}$ and a quorum set $D \in \mathcal{D}$, $\text{qset}(v, D)$ maps to the set of all valid quorum slices for v that satisfy D , i.e., $\text{qset}(v, D) : \mathbf{V} \times \mathcal{D} \rightarrow 2^{2^{\mathbf{V}}} := \{q \subseteq \mathbf{V} \mid v \in q \wedge q \text{ satisfies } D\}$.

Via the qset notation, quorum sets and quorum slices become equivalent representations that can be transformed into one another. A straightforward (but generally not space-efficient) way to express any k quorum slices $\{q_i \in 2^{\mathbf{V}} \mid i \in [0, k], v \in q_i$ of a node $v \in \mathbf{V}$ via a quorum set is $\text{qset}(v, (\emptyset, \mathcal{I}, 1))$, with $\mathcal{I} = \{(q_i, \emptyset, |q_i|) \mid i \in [0, k)\}$. Quorum sets are translated to quorum slices (values of \mathbf{Q}) by applying the qset function. For example (with $\mathbf{V} = \{0, 1, 2\}$):

$$\begin{aligned} \mathbf{Q}(0) &= \text{qset}(0, (\{1, 2\}, \emptyset, 1)) = \{\{0, 1\}, \{0, 2\}, \{0, 1, 2\}\} \\ \mathbf{Q}(1) &= \text{qset}(1, (\{0, 2\}, \emptyset, 2)) = \{\{0, 1, 2\}\} \\ \mathbf{Q}(2) &= \text{qset}(2, (\{0, 1, 2\}, \emptyset, 2)) = \{\{0, 2\}, \{1, 2\}, \{0, 1, 2\}\} \end{aligned}$$

In the above example, $\mathbf{V} = \{0, 1, 2\}$ and their quorum sets (as per \mathbf{Q}) form the FBAS (\mathbf{V}, \mathbf{Q}) . As a way to visualize (\mathbf{V}, \mathbf{Q}) , it can heuristically be represented as a graph where the existence of an edge (v_i, v_j) implies that v_j is included in at least one of v_i 's quorum slices:



3.3 Preconditions to liveness

A consensus system is *live* if it can externalize new values³. A consensus system built upon an FBAS is live if the FBAS contains an intact *quorum*— a group of FBAS nodes that can externalize new values by itself.

³ We content ourselves with a weak notion of liveness whereby a system is live as long as it is *non-blocking* [9] for one or more non-faulty nodes, i.e., as long as an execution path *exists* that allows one or more non-faulty nodes to make progress. This can also be called *plausible liveness*.

Definition 3.4 (*quorum* [16]) A set of nodes $U \subseteq \mathbf{V}$ in FBAS (\mathbf{V}, \mathbf{Q}) is a quorum iff $U \neq \emptyset$ and U contains a quorum slice for each member—i.e., $\forall v \in U \exists q \in \mathbf{Q}(v) : q \subseteq U$.

This is equivalent to stating that U satisfies the quorum sets of all $v \in U$. Quorums are therefore determined by the sum of all individual quorum set configurations. Continuing the previous example with nodes $\mathbf{V} = \{0, 1, 2\}$, we get the quorums $\mathcal{U} = \{\{0, 2\}, \{0, 1, 2\}\}$. We capture part of the semantics behind quorums by defining what it means for a consensus protocol to *honor* a given FBAS —namely that whenever values are externalized for a slot, at least one quorum of nodes must eventually externalize values as well.

Definition 3.5 (*protocol that honors an FBAS*) Let (\mathbf{V}, \mathbf{Q}) be an FBAS such that \mathbf{V} contains only non-faulty nodes, P a consensus protocol, and $N_i \subseteq \mathbf{V}$ the set of all nodes that, following P , eventually externalize a value for a given slot i . We say that P *honors* (\mathbf{V}, \mathbf{Q}) iff any nonempty N_i contains a quorum, i.e., $\forall i : N_i \neq \emptyset \vee \exists U \subseteq N$ such that U is a quorum for (\mathbf{V}, \mathbf{Q}) .

We say that (\mathbf{V}, \mathbf{Q}) has *quorum availability despite faulty nodes* iff there exists a $U \subseteq \mathbf{V}$ that is a quorum in (\mathbf{V}, \mathbf{Q}) and consists of only non-faulty nodes. Quorum availability despite faulty nodes is a necessary condition to achieving *liveness* in an FBAS, i.e., ensuring that non-faulty nodes can externalize new values independently of the behavior of faulty nodes [16].

Theorem 3.1 (*quorum availability \Leftarrow liveness*) Let (\mathbf{V}, \mathbf{Q}) be an FBAS and P a consensus protocol that honors (\mathbf{V}, \mathbf{Q}) . If P can provide liveness for (\mathbf{V}, \mathbf{Q}) independently of the behavior of faulty nodes, then (\mathbf{V}, \mathbf{Q}) enjoys quorum availability despite faulty nodes.

Proof Let $F \subseteq \mathbf{V}$ be the set of all faulty nodes and $(\mathbf{V} \setminus F, \mathbf{Q}')$ a sub-FBAS that contains all non-faulty nodes, with $\mathbf{Q}'(v) := \{q \in \mathbf{Q}(v) \mid q \subseteq \mathbf{V} \setminus F\}$ for $\forall v \in \mathbf{V} \setminus F$. P honors (\mathbf{V}, \mathbf{Q}) and can provide liveness independently of the behavior of nodes in F , therefore there must exist a protocol P' that can provide liveness while honoring $(\mathbf{V} \setminus F, \mathbf{Q}')$. Based on Def. 3.5, there is therefore at least one $U \subseteq \mathbf{V} \setminus F$ that is a quorum for $(\mathbf{V} \setminus F, \mathbf{Q}')$. U is, trivially, also a quorum for (\mathbf{V}, \mathbf{Q}) . \square

Given quorum availability despite faulty nodes, protocols like SCP can provide liveness [16]. In the case of SCP, this was previously demonstrated through correctness proofs [9] as well as formal verification and practical deployment experience [13]. Additional conditions to achieving liveness include the reaction (via quorum set adaptations, i.e., changes to \mathbf{Q}) to (detectable) timing attacks [13]. We defer to works such as [2,3,14,16] for an in-depth exploration of the mechanics and guarantees of consensus protocols for the FBAS setting.

3.4 Preconditions to safety

A set of nodes in an FBAS enjoy *safety* if no two of them ever externalize different values for the same slot [16]. In a blockchain context, a lack of safety guarantees translates into the possibility of forks and double spends. Protocols that honor an FBAS can only guarantee safety if the FBAS enjoys *quorum intersection*.

Definition 3.6 (*quorum intersection* [16]) A given FBAS enjoys quorum intersection iff any two of its quorums share a node—i.e., for all quorums U_1 and U_2 , $U_1 \cap U_2 \neq \emptyset$.

For example, the set of quorums $\{\{0, 2\}, \{0, 1, 2\}\}$ intersects, whereas introducing an additional quorum $\{1, 4\}$ would break quorum intersection. In the latter scenario, $\{0, 2\}$ and $\{1, 4\}$ could induce two new, separated FBASs [14]. We say that an FBAS enjoys *quorum intersection despite faulty nodes* if every two quorums that contain non-faulty nodes intersect in at least one non-faulty node, even if all faulty nodes change their quorum sets in arbitrary ways or report different quorum sets to different peers. Formally, quorum intersection despite faulty nodes is defined via a *delete* operation that transforms an FBAS based on the assumption that a given set of nodes is acting in the most harmful (to safety) way possible.

Definition 3.7 (*delete* [16]) If (\mathbf{V}, \mathbf{Q}) is an FBAS and $F \subseteq \mathbf{V}$ a set of nodes, then to *delete* F from (\mathbf{V}, \mathbf{Q}) , written $(\mathbf{V}, \mathbf{Q})^F$, means to compute the modified FBAS $(\mathbf{V} \setminus F, \mathbf{Q}^F)$ where $\mathbf{Q}^F(v) = \{q \setminus F, q \in \mathbf{Q}(v)\}$.

If $F \subseteq \mathbf{V}$ is the set of all faulty nodes, then an FBAS (\mathbf{V}, \mathbf{Q}) enjoys quorums intersection despite faulty nodes iff $(\mathbf{V}, \mathbf{Q})^F$ enjoys quorum intersection. If quorum intersection despite faulty nodes is not given, safety cannot be guaranteed (although it can be maintained by chance).

Theorem 3.2 (*quorum intersection \iff guaranteed safety*)

Let (\mathbf{V}, \mathbf{Q}) be an FBAS and P a consensus protocol that can provide liveness for any FBAS with quorum availability despite faulty nodes, while honoring the respective FBAS. Let P furthermore be non-trivial, in the sense that externalized values are non-deterministic and depend on user input. If P can guarantee safety for all non-faulty nodes in \mathbf{V} , then (\mathbf{V}, \mathbf{Q}) enjoys quorum intersection despite faulty nodes.

Proof Let $F \subseteq \mathbf{V}$ be the set of all faulty nodes and $(\mathbf{V}', \mathbf{Q}') := (\mathbf{V}, \mathbf{Q})^F$. If $(\mathbf{V}', \mathbf{Q}')$ does not enjoy quorum intersection, then there are two quorums $U_1, U_2 \subset \mathbf{V}'$ so that $U_1 \cap U_2 = \emptyset$. For $i \in \{1, 2\}$, let Q_i be defined such that $\forall v \in U_i : Q_i(v) := \{q \in \mathbf{Q}'(v) \mid q \subseteq U_i\}$. Then both (U_1, Q_1) and (U_2, Q_2) form FBASs with quorum availability. As P can provide liveness for any FBAS with quorum availability,

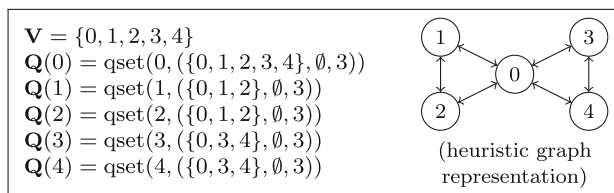


Fig. 1 Example FBAS (\mathbf{V}, \mathbf{Q})

(U_1, Q_1) and (U_2, Q_2) can externalize values for the same slots without any communication taking place between nodes in U_1 and nodes in U_2 .

As P is non-trivial, the externalized values can differ, i.e., safety cannot be guaranteed. \square

As formally proven by García-Pérez and Gotsman [8], an FBAS that enjoys quorum intersection induces a Byzantine quorum system [15], and an FBAS that enjoys quorum intersection despite faulty nodes can induce a dissemination quorum system [15]. These results are independent of attempts by faulty nodes to lie about their quorum set configuration [8]. There is strong evidence that protocols like SCP can guarantee safety in any FBAS with quorum intersection despite faulty nodes [2,9,13,14].

4 Concepts for further analysis

In the following, we define new concepts for capturing relevant properties of concrete FBAS instances. While it is typical in the BFT literature to construct proofs based on *assuming* which sets of nodes *can* fail simultaneously (i.e., which are the *fail-prone sets* [15]), we instead investigate which sets of nodes *have to* fail in order for global liveness and safety guarantees to become void. This perspective uncovers the liveness and safety *buffers* a given (potentially non-trivial) quorum system has and is thus highly relevant for the monitoring and evaluation of systems deployed in practice. While defined based on the FBAS model, the proposed concepts are readily transferable to more general quorum system formalizations (e.g., recall that safety-enabling FBASs induce Byzantine quorum systems [8]).

For illustration, we will be using the example FBAS defined via Fig. 1. An analysis of a slightly larger example FBAS is presented in Appendix B. Appendix A contains formal write-ups and proofs of various corollaries and theorems relevant to this section.

4.1 Starting point: Minimal quorums

As a prerequisite to subsequent analyses, it is helpful to understand which *quorums* (cf. Def. 3.4) exist in an FBAS. We will be focusing on *minimal quorums*, i.e., quorums

$\hat{U} \subseteq \mathbf{V}$ for which there is no proper subset $U \subset \hat{U}$ that is also a quorum. Informally, the set of all minimal quorums $\hat{\mathcal{U}}$ carries sufficient information for precisely determining FBAS-wide liveness properties, while being of significantly smaller size than the set of *all* quorums \mathcal{U} .

Definition 4.1 (*minimal node set*) Within the set of node sets $\mathcal{N} \subseteq 2^{\mathbf{V}}$, a member set $\hat{N} \in \mathcal{N}$ is minimal iff none of its proper subsets is included in \mathcal{N} —i.e., $\forall N \in \mathcal{N}, N \not\subseteq \hat{N}$.

The FBAS depicted in Fig. 1 has the quorums $\mathcal{U} = \{\{0, 1, 2\}, \{0, 3, 4\}, \{0, 1, 2, 3, 4\}\}$ and consequently the minimal quorums $\hat{\mathcal{U}} = \{\{0, 1, 2\}, \{0, 3, 4\}\}$.

The notion of minimal quorums is helpful, among other things, for efficiently determining whether an FBAS enjoys quorum intersection [12]: it can be shown that an FBAS enjoys quorum intersection iff every two of its minimal quorums intersect (Cor. A.1).

4.2 Minimal blocking sets

As per Thm. 3.1, an FBAS (\mathbf{V}, \mathbf{Q}) cannot enjoy liveness if it doesn't contain at least one non-faulty quorum. Considering the state of the art in consensus protocols for the FBAS setting and their formal verification (s.a. Sec. 3.3), quorum availability despite faulty nodes is furthermore the *only* precondition to achieving liveness that depends on (\mathbf{V}, \mathbf{Q}) and arguably the most difficult to satisfy in a practical deployment. However, while quorum availability can easily be checked based on \mathbf{Q} , faulty nodes are usually not readily identifiable as such in practice. We therefore propose, as a means to grasping liveness risks, to look at sets of nodes that, if faulty, can undermine quorum availability.

Definition 4.2 (*blocking set*) Let $\mathcal{U} \subseteq 2^{\mathbf{V}}$ be the set of all quorums of the FBAS (\mathbf{V}, \mathbf{Q}) . We denote the set $B \subseteq \mathbf{V}$ as *blocking* iff it intersects every quorum of the FBAS—i.e., $\forall U \in \mathcal{U}, B \cap U \neq \emptyset$

For example: $\{0\}$ and $\{1, 3\}$ are both blocking sets for $\mathcal{U} = \{\{0, 1, 2\}, \{0, 3, 4\}, \{0, 1, 2, 3, 4\}\}$.

Corollary 4.1 (**blocking sets and liveness**) *Control over any blocking set B is sufficient for compromising the liveness of an FBAS (\mathbf{V}, \mathbf{Q}) .*

Proof As B intersects all quorums of the FBAS, there is no quorum that can be formed without cooperation by B . Without at least one non-faulty quorum, liveness is not possible as per Thm 3.1. \square

Notably, blocking sets can also block liveness *selectively*, enabling *ensorship*. As nodes from the blocking set are present in every quorum, consensus will never be reached on any value that the blocking set opposes to. For example,

in the context of Stellar, the blocking set could block the ratification of transactions involving specific accounts. We chose the term blocking in analogy to the *v-blocking sets* introduced in [16]. As an important distinction, we use the term *blocking set* to refer to a property of the whole FBAS (\mathbf{V}, \mathbf{Q}) , as opposed to a property of an individual node $v \in \mathbf{V}$.

In the above example, $\{0\}$ and $\{1, 3\}$ are not only blocking sets with respect to \mathcal{U} , they are *minimal blocking sets*, i.e., none of their proper subsets is a blocking set⁴. In essence, minimal blocking sets describe minimal threat (respectively, fail) scenarios w.r.t. liveness.

4.3 Minimal splitting sets

As per Thm. 3.2, an FBAS can only be considered safe (as one coherent system) as long as it enjoys quorum intersection despite faulty nodes, i.e., as long as each two of its quorums intersect even after all faulty nodes have been deleted (as per Def. 3.7). For practical purposes, quorum intersection despite faulty nodes is furthermore a *sufficient* condition for achieving safety in an FBAS, considering protocols like SCP and the correctness proofs surrounding them (s.a. Sec. 3.4). Hence, for assessing the risk to safety, it is interesting to identify sets of nodes that can cause an FBAS to effectively lose quorum intersection. We call such a set of nodes a *splitting set*, as it can, if faulty, cause at least two quorums to diverge, splitting the FBAS.

Definition 4.3 (*splitting set*) We denote the set $S \subseteq \mathbf{V}$ a splitting set iff $(\mathbf{V}, \mathbf{Q})^S$ lacks quorum intersection—i.e., there are distinct quorums U_1 and U_2 of $(\mathbf{V}, \mathbf{Q})^S$ so that $U_1 \cap U_2 = \emptyset$.

In the above example with $\hat{\mathcal{U}} = \{\{0, 1, 2\}, \{0, 3, 4\}\}$, $\{0\}$ is already a splitting set, as $(\mathbf{V}, \mathbf{Q})^{\{0\}}$ induces the two non-intersecting quorums $\{1, 2\}$ and $\{3, 4\}$. Intuitively, $\{0\}$ is a splitting set of (\mathbf{V}, \mathbf{Q}) because it forms the intersection of the quorums $\{0, 1, 2\}$ and $\{0, 3, 4\}$.

The existence of a faulty splitting set violates quorum intersection despite faulty nodes and therefore, as per Thm. 3.2, threatens safety. Informally, the members of a splitting set can perform two types of actions to compromise safety in practice (s.a. Thm. A.1). On the one hand, they can change their quorum configurations (or lie about them) to cause existing quorums to shrink or new quorums to emerge, both with the goal of reducing the overlap between quorums. On the other hand, whenever the intersection of two (minimal) quorums is comprised entirely of faulty nodes, these nodes can agree to different statements in each quorum, causing the quorums to externalize conflicting values and in this way diverge.

⁴ For completeness, the set of all minimal blocking sets w.r.t. \mathcal{U} is $\hat{\mathcal{B}} = \{\{0\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}\}$.

As with blocking sets, we are especially interested in finding the *minimal splitting sets* $\hat{\mathcal{S}} \subset 2^{\mathbf{V}}$ of an FBAS⁵ (\mathbf{V}, \mathbf{Q}) . Minimal splitting sets describe minimal threat scenarios w.r.t. safety.

4.4 Top tier

For narrowing down notions of “centralization” with respect to FBASs, we propose the concept of a *top tier*. Informally, the top tier is the set of nodes in the FBAS that is exclusively relevant when determining minimal blocking sets and hence the liveness buffers of an FBAS.

Definition 4.4 (*top tier*) The top tier of an FBAS (\mathbf{V}, \mathbf{Q}) is the set of all nodes that are contained in one or more minimal quorums—i.e., if $\hat{\mathcal{U}} \subseteq 2^{\mathbf{V}}$ is the set of all minimal quorums of the FBAS, $T = \bigcup \hat{\mathcal{U}}$ is its top tier.

In the above example, it in fact holds that $T = \{0, 1, 2, 3, 4\} = \mathbf{V}$.

It can be shown that each minimal blocking set consists exclusively of top tier nodes (Cor. A.5), and each top tier node is included in at least one minimal blocking set (Thm. A.2). The FBAS (\mathbf{V}, \mathbf{Q}) with top tier T has therefore the same properties w.r.t. global liveness as the FBAS induced by T , i.e., the FBAS (T, \mathbf{Q}') with $\mathbf{Q}'(v) := \{q \cap T \mid q \in \mathbf{Q}(v)\}$.

This observation has direct implications for the computational complexity of FBAS analysis (further discussed in Sec. 5), and for the performance of FBAS-based consensus protocols. A consensus round in SCP (the so far only production-ready protocol for the FBAS setting, to the best of our knowledge) can demonstrably be completed in $O(|T|^2)$ messages. While classical consensus protocols with quadratic message complexity (such as PBFT [4]) are notorious for becoming unusable in larger validator groups, several improved protocols have recently emerged that target the blockchain use case and scenarios with 100 and more validators [20,22]. As a possible avenue for future exploration—for FBASs with a *symmetric top tier*, existing permissioned protocols could be adapted without much modification.

Definition 4.5 (*symmetric top tier*) The top tier T of an FBAS (\mathbf{V}, \mathbf{Q}) is a symmetric top tier iff all top tier nodes have identical quorum sets—i.e., $\exists D \in \mathcal{D}, \forall v \in T : \mathbf{Q}(v) = \text{qset}(v, D)$.

Symmetric top tiers are also significantly more amenable to analysis. For example, in FBASs with a symmetric top tier T and a non-nested top tier quorum set (T, \emptyset, t) , it holds that any minimal blocking set has cardinality $|\hat{B}| = |T| - t + 1$ (Thm. A.3) and any minimal splitting set that can cause two top tier nodes to diverge from each other has cardinality $|\hat{S}| = 2t - |T|$ (Thm. A.4).

⁵ In the above example, $\{0\}$ is the only minimal splitting set w.r.t. \mathcal{U} , i.e., the set of all minimal splitting sets is $\hat{\mathcal{S}} = \{\{0\}\}$.

5 Analysis algorithms

In the following, we propose algorithms for performing the analyses introduced in Sec. 4. We describe them as pseudocode that necessarily abstracts away some implementation details and optimizations. As a companion to this paper, we release a well-tested implementation of the presented algorithms as open source (`fbas_analyzer`⁶). After outlining algorithms for enumerating minimal quorums (foundation for further analyses), determining quorum intersection (necessary condition for safety), enumerating minimal blocking sets (liveness “buffers”), enumerating minimal splitting sets (safety “buffers”), and efficiently dealing with symmetric top tiers, the section concludes with a short empirical study on analysis scalability.

5.1 Minimal quorums

Algorithm 1 describes a branch-and-bound algorithm for finding all minimal quorums. It is based on a quorum enumeration procedure originally described in [12]. Previous algorithms did not rigorously filter out non-minimal quorums, which we realize through `is_minimal_quorum`. The set of all minimal quorums of an FBAS defines its top tier (cf. Sec. 4.4) and can be used for determining whether the FBAS enjoys quorum intersection.

The keystone of the algorithm is the function `fmq_step` that takes a current quorum *candidate* U , a sorted list of *yet-to-be-considered nodes* V and a reference to \mathbf{Q} for mapping nodes to their quorum sets. The algorithm implements a classical branching pattern: at each invocation of `fmq_step` in which U is not already a quorum, the next node in V is taken out and, in one branch, added to U , and, in the other, not. Hopeless branches are identified early using the `is_satisfiable` function.

As proposed in [12], we initially sort V using a heuristic such as PageRank [19] which can improve the algorithm’s performance in practice. Another important optimization from [12], that we leave out in our pseudocode for greater clarity, is the partitioning of \mathbf{V} into *strongly connected components*⁷ so that `find_minimal_quorums` must be applied only to (often significantly smaller) subsets of \mathbf{V} . Tarjan [21] gives an algorithm for performing this preprocessing step in linear time.

As noted in other works (e.g., [1,12]), determining quorum intersection, and hence also enumerating all min-

⁶ https://github.com/wiberlin/fbas_analyzer; Our Rust-based library has been integrated into <https://stellarbeat.io/> (a popular monitoring service for the Stellar network) and supports in-browser usage—cf. our interactive analysis website at https://trudi.weizenbaum-institut.de/stellar_analysis/.

⁷ Based on the heuristic representation of the FBAS as a directed graph.

Algorithm 1: Find minimal quorums

```

1 Function find_minimal_quorums( $(\mathbf{V}, \mathbf{Q})$ ):
   Data: An FBAS  $(\mathbf{V}, \mathbf{Q})$ .
   Result:  $\hat{U}$ , the set of all minimal quorums of  $(\mathbf{V}, \mathbf{Q})$ .
2    $V \leftarrow \mathbf{V}$  sorted by, e.g., PageRank [19] (cf. [12]);
3   return fmq_step( $\emptyset, V, \mathbf{Q}$ );
4 Function fmq_step( $U, V, \mathbf{Q}$ ):
5   if is_quorum( $U, \mathbf{Q}$ ) then
6     if is_minimal_for_quorum( $U, \mathbf{Q}$ ) then
7       return  $\{U\}$ ;
8     else return  $\emptyset$ ;
9   else if is_satisfiable( $U, V, \mathbf{Q}$ ) then
10     $v \leftarrow$  next in  $V$ ;
11    return fmq_step( $U \cup \{v\}, V \setminus \{v\}, \mathbf{Q}$ )  $\cup$ 
      fmq_step( $U, V \setminus \{v\}, \mathbf{Q}$ );
12  else return  $\emptyset$ ;
13 Function is_quorum( $U, \mathbf{Q}$ ):
14  return  $\forall v \in U \exists q \in \mathbf{Q}(v) : q \subseteq U$ ;
15 Function is_satisfiable( $U, V, \mathbf{Q}$ ):
16  return  $\forall v \in U \exists q \in \mathbf{Q}(v) : q \subseteq U \cup V$ ;
17 Function is_minimal_for_quorum( $U, \mathbf{Q}$ ):
18  for  $v \in U$  do
19    if contains_quorum( $U \setminus \{v\}, \mathbf{Q}$ ) then
20      return false;
21  end
22  return true;
23 Function contains_quorum( $U, \mathbf{Q}$ ):
24  // remove non-satisfiable nodes
25  while  $\exists v \in U \forall q \in \mathbf{Q}(v) : q \not\subseteq U$  do
26     $U \leftarrow \{v \in U \mid \exists q \in \mathbf{Q}(v) : q \subseteq U\}$ ;
27  end
28  return  $U \neq \emptyset$ ;

```

imal quorums, is NP-hard. Consequently, our algorithm has exponential time complexity. For an FBAS with $n = |\mathbf{V}|$ nodes and a top tier of size $m = |T|$ we find all $k \leq \binom{m}{\lfloor \frac{m}{2} \rfloor}$ minimal quorums in $O(2^m)$. Note that in practice the number of de-facto considered nodes n is greatly reduced through polynomial-time preprocessing steps such as strongly-connected-component analysis and heuristics-based sorting, yielding actual running times that are close to the $O(2^m)$ bound.

5.2 Quorum intersection

Quorum intersection is a central property for being able to guarantee safety in an FBAS (cf. Sec. 4.3). Quorum intersection can be determined by checking the pairwise intersection of all minimal quorums (Cor. A.1). This straightforward approach, that was also proposed in [12], is embodied in Algorithm 2.

In this paper, we propose an additional, alternative algorithm (Algorithm 3), that doesn't check for pairwise intersections but instead checks whether the *complement sets* of found quorums contain quorums themselves. If this is

Algorithm 2:

 Checking for quorum intersection via approach from [12].

```

1 Function has_quorum_intersection( $(\mathbf{V}, \mathbf{Q})$ ):
   Data: An FBAS  $(\mathbf{V}, \mathbf{Q})$ .
   Result: true if the FBAS enjoys quorum intersection, false
     else.
2    $\hat{U} \leftarrow$  find_minimal_quorums( $(\mathbf{V}, \mathbf{Q})$ );
3   return  $\forall \hat{U}_i, \hat{U}_j \in \hat{U} : \hat{U}_i \cap \hat{U}_j \neq \emptyset$ ;

```

never the case, the FBAS enjoys quorum intersection. This approach for checking for quorum intersection has the benefit that only a constant number of node sets must be held in memory at the same time, as opposed to all minimal quorum sets as in Algorithm 2. The space complexity of the check is therefore reduced from exponential to linear.

Algorithm 3:

 Checking for quorum intersection via alternative approach with linear space complexity.

```

1 Function has_quorum_intersection( $(\mathbf{V}, \mathbf{Q})$ ):
   Data: An FBAS  $(\mathbf{V}, \mathbf{Q})$ .
   Result: true if the FBAS enjoys quorum intersection, false
     else.
2   for  $\hat{U} \in$  find_minimal_quorums( $(\mathbf{V}, \mathbf{Q})$ ) do
3     if contains_quorum( $\mathbf{V} \setminus \hat{U}$ ) then
4       return false;
5   end
6   return true;

```

Our implementation of Algorithm 3 is also empirically faster for many FBASs, probably because contains_quorum scales better than iterating once over all minimal quorums, and because less data must be written to memory. For both algorithms, we leave out optimization details such as leveraging the fact that quorum intersection is guaranteed to hold if all minimal quorums $\hat{U} \in \hat{U}$ have cardinality greater than $\frac{|\cup \hat{U}|}{2}$. In Algorithm 3, for example, it suffices to check only minimal quorums with fewer than $\frac{|\cup \hat{U}|}{2}$ members.

5.3 Minimal blocking sets

Algorithm 4 presents our algorithm for enumerating all minimal blocking sets based on a branch-and-bound strategy. The check whether a given *candidate set* B is blocking is performed by checking whether the FBAS contains any quorums after B is removed from the node population. If a blocking set can still be formed from B and the yet-to-be-considered nodes V (this is the pruning rule), the enumeration continues, branching via either adding the next node in V to the candidate set or discarding it altogether. The order in which nodes are visited can be tuned using a suitable heuristic—we sort nodes using PageRank [19] (as for finding minimal quorums)

in the example pseudocode and our current implementation. Like for Algorithm 1, the complexity of Algorithm 4 is in $O(2^n)$ (for an FBAS with n nodes) with a likely practical average case complexity of $O(2^m)$ (m being the size of the top tier).

Algorithm 4: Find minimal blocking sets

```

1 Function find_minimal_blocking_sets( $(\mathbf{V}, \mathbf{Q})$ ):
   Data: An FBAS  $(\mathbf{V}, \mathbf{Q})$ .
   Result:  $\hat{B}$ , the set of all minimal blocking sets of  $(\mathbf{V}, \mathbf{Q})$ .
2    $V \leftarrow \mathbf{V}$  sorted by, e.g., PageRank [19];
3   return fmb_step( $\emptyset, V, \mathbf{Q}$ );
4 Function fmb_step( $B, V, \mathbf{Q}$ ):
5   if is_blocking( $B, V, \mathbf{Q}$ ) then
6     if is_minimal_for_blocking( $B, V, \mathbf{Q}$ ) then
7       return  $\{B\}$ ;
8     else return  $\emptyset$ ;
9   else if is_blocking( $B \cup V, V, \mathbf{Q}$ ) then
10     $v \leftarrow$  next in  $V$ ;
11    return fmb_step( $B \cup \{v\}, V \setminus \{v\}, \mathbf{Q}$ )  $\cup$ 
        fmb_step( $B, V \setminus \{v\}, \mathbf{Q}$ );
12  else return  $\emptyset$ ;
13 Function is_blocking( $B, V, \mathbf{Q}$ ):
14  return  $\neg$ contains_quorum( $V \setminus B, \mathbf{Q}$ );
15 Function is_minimal_for_blocking( $B, V, \mathbf{Q}$ ):
16  for  $v \in B$  do
17    if is_blocking( $B \setminus \{v\}, V, \mathbf{Q}$ ) then
18      return false;
19  end
20  return true;

```

5.4 Minimal splitting sets

Algorithm 5 presents our algorithm for enumerating all minimal splitting sets. We again perform a branch-and-bound search. The final condition for accepting a *candidate set* S is whether deleting it (cf. Def. 3.7) from the FBAS causes the FBAS to lose quorum intersection.

This check is significantly more expensive than the corresponding checks in Algorithm 1 and Algorithm 4. Additionally, unlike the previously presented algorithms, Algorithm 5 also needs to consider non-top tier nodes as candidates. We incorporate the observation (from Thm. A.1) that a node can only be part of a minimal splitting set if it is part of a minimal quorum (only then can it be part of an intersection of minimal quorums) or if a change of its quorum set can potentially cause new, smaller quorums to emerge. Consequently, we consider as candidates all top tier nodes and all nodes that are *quorum expanders*: nodes that are part of a quorum slice of another node that is not a quorum slice for themselves (formal definition in Def. A.1). Informally, by not sharing a quorum slice with a node they affect, quorum expanders may force quorums to expand beyond this quorum slice. By

changing their quorum set, quorum expanders could reverse this effect, leading to smaller quorums and, accordingly, an increased risk to quorum intersection.

The `has_potential` function embodies an explicit pruning condition for the branch-and-bound search. Here, we check whether a change in the FBAS’s minimal quorums is possible if some or all outstanding candidate nodes V are joined with the current candidate set S . As a heuristic to avoid actually calculating minimal quorums, we check whether the quorum-containing strongly connected components of the FBAS change after deleting V in addition to S .

For improving readability and comprehension, we leave out various details and smaller optimizations from our pseudocode listing for Algorithm 5. Among other things, we don’t include our full algorithms for enumerating `quorum_expanders` and deliberately ignore opportunities for caching and reusing the results of costly operations.

Algorithm 5: Find minimal splitting sets

```

1 Function find_minimal_splitting_sets( $(\mathbf{V}, \mathbf{Q})$ ):
   Data: An FBAS  $(\mathbf{V}, \mathbf{Q})$ .
   Result:  $\hat{S}$ , the set of all minimal splitting sets of  $(\mathbf{V}, \mathbf{Q})$ .
2    $V \leftarrow \bigcup$  find_minimal_quorums( $(\mathbf{V}, \mathbf{Q})$ );
3    $V \leftarrow V \cup$  quorum_expanders( $(\mathbf{V}, \mathbf{Q})$ );
4    $V \leftarrow V$  sorted by, e.g., number of affected nodes;
5    $A \leftarrow \mathbf{V}$ ;
6    $S \leftarrow$  fs_step( $\emptyset, V, A, (\mathbf{V}, \mathbf{Q})$ );
7   return reduce_to_minimal_sets( $S$ );
8 Function fs_step( $S, V, (\mathbf{V}, \mathbf{Q})$ ):
9   if  $\neg$ has_quorum_intersection( $(\mathbf{V}, \mathbf{Q})^S$ ) then
10    return  $\{S\}$ ;
11  else if has_potential( $S, V, (\mathbf{V}, \mathbf{Q})$ ) then
12     $v \leftarrow$  next in  $V$ ;
13    return fs_step( $S \cup \{v\}, V \setminus \{v\}, (\mathbf{V}, \mathbf{Q})$ )  $\cup$ 
        fs_step( $S, V \setminus \{v\}, (\mathbf{V}, \mathbf{Q})$ );
14  else return  $\emptyset$ ;
15
16 Function quorum_expanders( $(\mathbf{V}, \mathbf{Q})$ ):
17  return
     $\{v \in \mathbf{V} \mid \exists v' \in \mathbf{V}, q' \in \mathbf{Q}(v') : v \in q' \wedge (\forall q \in \mathbf{Q}(v) : q \not\subseteq q')\}$ ;
18
19 Function has_potential( $S, V, (\mathbf{V}, \mathbf{Q})$ ):
20  return  $(\exists v \in V \exists q \in \mathbf{Q}(v) : q \subseteq S \cup V) \vee$ 
     $(\text{quorum\_clusters}((\mathbf{V}, \mathbf{Q})^{S \cup V}) \neq$ 
     $\text{quorum\_clusters}((\mathbf{V}, \mathbf{Q})^S))$ ;
21
22 Function quorum_clusters( $(\mathbf{V}, \mathbf{Q})$ ):
23   $\mathcal{N} \leftarrow$  strongly connected components of  $(\mathbf{V}, \mathbf{Q})$ ;
24  return  $\{N \in \mathcal{N} \mid \text{contains\_quorum}(N)\}$ ;
25
26 Function reduce_to_minimal_sets( $S$ ):
27  return  $\{\hat{S} \in S \mid \forall S \in S : S \not\subseteq \hat{S}\}$ ;

```

The asymptotic complexity of Algorithm 5 remains in $O(2^n)$, respectively $O(2^{|T \cup X|})$ where T is the top tier and X the set of all quorum expanders. However, due to the costly

acceptance check for splitting sets and the larger number of nodes that need to be considered, the algorithm is significantly slower than Algorithm 1 and Algorithm 4 in practice.

5.5 Symmetric clusters

As a generalization of symmetric top tiers (Def. 4.5), we define *symmetric clusters* of an FBAS (\mathbf{V}, \mathbf{Q}) as groups of nodes $Y \subseteq \mathbf{V}$ such that $\exists D \in \mathcal{D}, \forall v \in Y : \mathbf{Q}(v) = \text{qset}(v, D)$ and $\bigcup \{ \mathbf{Q}(v), v \in Y \} = Y$. If an FBAS has one symmetric cluster Y and $\mathbf{V} \setminus Y$ does not contain a quorum, Y is the symmetric top tier of (\mathbf{V}, \mathbf{Q}) ⁸.

Symmetric clusters can be found in polynomial time, by grouping nodes with identical quorum set configurations (values for \mathbf{Q}) and checking the above condition for each thus formed candidate set.

Symmetric clusters can be analyzed significantly more efficiently. For example, an FBAS with a non-nested symmetric top tier is isomorphic to a classical, threshold-based quorum system (s.a. Thm. A.3 and A.4). For symmetric clusters formed around a nested quorum set, minimal quorums and minimal blocking sets can be enumerated without the overhead of checking candidate sets, by recursively listing combinations and forming their Cartesian product. If the interest is to find only such splitting sets that can cause nodes within the symmetric cluster to diverge, then the same is true for minimal splitting sets.

5.6 Analysis performance

Our analysis approach requires the enumeration of minimal quorums, minimal blocking sets and minimal splitting sets—which in all three cases is an NP-hard problem. It is unclear, however, what this means for the practical limitations of thoroughly determining the safety and liveness buffers of an FBAS. Practical limitations are difficult to conclusively determine as the real-life performance of analyses depends heavily on the topology of analyzed FBASs and the implementation of the algorithms.

In the following, we present a short exploratory study into the scalability of our own implementation. We construct synthetic FBASs of increasing size that consist of only a top tier. In the first series of presented experiments (Fig. 2), we construct FBASs (\mathbf{V}, \mathbf{Q}) resembling classical $3f + 1$ quorum systems:

$$\forall v \in \mathbf{V} : \mathbf{Q}(v) = \text{qset}(v, (\mathbf{V}, \emptyset, \lceil \frac{2|\mathbf{V}| + 1}{3} \rceil))$$

In a second series of experiments (Fig. 3), we approximate the structure of the Stellar network’s top tier where each organi-

⁸ If an FBAS has $l > 1$ symmetric clusters or $\mathbf{V} \setminus Y$ does contain a quorum, (\mathbf{V}, \mathbf{Q}) does not enjoy quorum intersection.

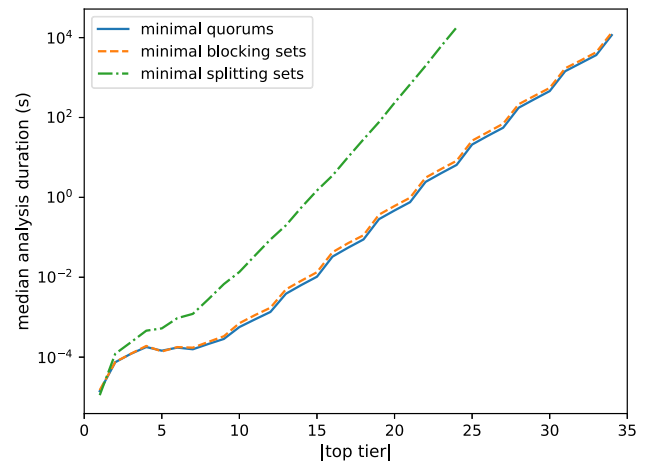


Fig. 2 Analysis duration for FBASs resembling classical $3f + 1$ quorum systems. Analysis optimizations for symmetric top tiers were turned off

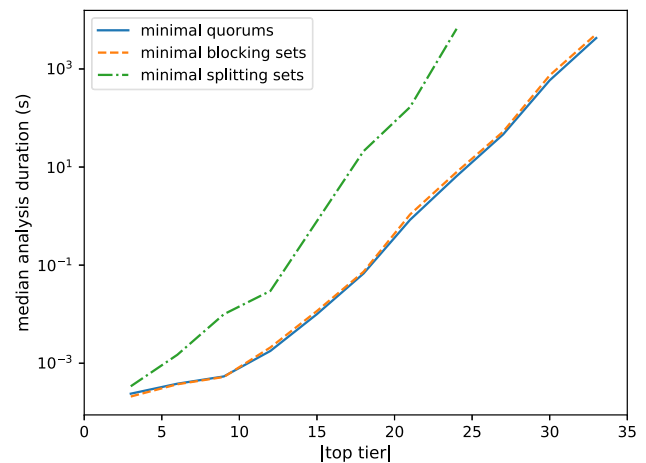


Fig. 3 Analysis duration for FBASs resembling the structure of the Stellar network top tier. Analysis optimizations for symmetric top tiers were turned off

zation is represented by (usually) 3 physical nodes arranged in crash failure-tolerating $2f + 1$ inner quorum sets:

$$\begin{aligned} \mathbf{V} &= \{v_0, v_1, \dots, v_{n-1}\}, n = 3m \\ \mathcal{I} &= \{(\{v_{3i}, v_{3i+1}, v_{3i+2}\}, \emptyset, 2) \mid i \in [0, m)\} \\ \forall v \in \mathbf{V} : \mathbf{Q}(v) &= \text{qset}(v, (\emptyset, \mathcal{I}, \lceil \frac{2m + 1}{3} \rceil)) \end{aligned}$$

We enumerate all minimal quorums, minimal blocking sets and minimal splitting sets of thus generated FBASs and record the time to completion of each of these operations. All analyses were single-threaded and performed on regular server-class hardware. We explicitly deactivated all optimizations based on detecting and exploiting symmetric clusters, so that the results of this study reflect the performance of the more expensive Algorithms 1, 4 and 5.

Figures 2 and 3 depict the median measured times on a log scale, from a set of 10 measurements per FBAS size (we performed the same analysis 10 times, recording individual times). As was expected, analysis durations raise exponentially with growing top tier sizes m . Analyses start requiring more than an hour to finish at $m \geq 23$ for flat symmetric top tiers and $m \geq 24$ for Stellar-like topologies. This is a cautiously positive result—top tier sizes observed in practice are currently in the range of 7 organizations (23 raw nodes) for the Stellar network (cf. Appendix C) and 7 organizations (10 raw nodes) for the MobileCoin network [18]. It is likely that, for example through parallelization or the development of additional optimizations for “almost symmetric” FBASs, the analysis durations for naturally occurring FBASs can be reduced further.

6 Bootstrapping FBASs

The reported openness enabled through the FBAS paradigm comes at the cost of increased configuration responsibilities for node operators. As discussed in Sec. 3, each node must become associated with a quorum set (respectively quorum slices) in order to become a useful part of an FBAS. We will refer to this process as *quorum set configuration* (QSC). But how should a node operator go about QSC? Based on the analytical toolset introduced in Sec. 4, we can now investigate what kinds of *QSC policies* are plausible and in what kind of FBASs they result.

Notably, we explore how *individual preferences* (such as which nodes should be “trusted”) can be mapped to the quorum set formalism. Based on experiments that use Internet topology as a representative graph representation of interdependence and trust, we conclude that purely individualistic configuration policies can result in systems with low liveness and high complexity. We outline possible directions for future research by sketching policies with a *strategic* element and empirically demonstrating their effectiveness.

6.1 QSC policies and their evaluation

A QSC policy is individually and repeatedly invoked for each node $v \in \mathbf{V}$. It takes information about a current FBAS instance (\mathbf{V}, \mathbf{Q}) as input and returns a quorum set for v , setting a new value for $\mathbf{Q}(v)$. We use the quorum set formalization introduced in Sec. 3.2. For illustration, consider the following trivial policy:

$$\forall v \in \mathbf{V} : \mathbf{Q}(v) = \text{qset}(v, (\mathbf{V}, \emptyset, |\mathbf{V}|)) \quad (\text{Super Safe QSC})$$

If implemented by all nodes in \mathbf{V} , **Super Safe QSC** leads to each node having only one quorum slice— \mathbf{V} itself ($\mathbf{Q}(v) = \{\mathbf{V}\}$). The policy maximizes safety but leads to blocking sets

of cardinality 1—any node can block the single quorum in the induced FBAS.

As an improvement, the threshold of the formed quorum sets can be set in resemblance to classical BFT protocols:

$$\forall v \in \mathbf{V} : \mathbf{Q}(v) = \text{qset}(v, (\mathbf{V}, \emptyset, \lceil \frac{2|\mathbf{V}| + 1}{3} \rceil)) \quad (\text{Ideal Open QSC})$$

For $|\mathbf{V}| = 3f + 1$ with an $f \in \mathbb{Z}^+$, setting the threshold to $t = \lceil \frac{2|\mathbf{V}| + 1}{3} \rceil$ leads to FBASs in which any $2f + 1$ nodes form a (minimal) quorum. This results in both all minimal blocking sets and all minimal splitting sets of the induced FBAS having cardinality $f + 1$, i.e., both safety *and* liveness can be maintained in the face of up to f node failures.

6.1.1 Choosing validators

The preceding example policies construct non-nested quorum sets that use as validators U the set of all nodes in the FBAS ($U = \mathbf{V}$). These are clearly toy examples—if anything else, without additional mechanisms to restrict or filter the membership in \mathbf{V} , \mathbf{V} can easily become dominated by faulty Sybil [5] nodes.

In the scope of this work, and in line with the motivation behind the FBAS paradigm, we consider \mathbf{V} to enjoy *open membership*, with no universally trusted whitelist or ranking. For arriving at sensible choices for U , QSC policies must therefore take individual knowledge into account.

6.1.2 Modeling individual preferences

QSC policies based on individual preferences contribute node-local knowledge to the collective FBAS configuration. For example:

- Which nodes are *trusted* to be (and stay) non-faulty. It is often implied that QSC should reflect some form of trust, e.g., in wordings such as “flexible trust” [16] or “asymmetric distributed trust” [2]. While reasoning about the future behavior of participants in a consensus protocol might be an overwhelming task for node operators, they may at least encode plausible beliefs about non-Sybilness [5] (i.e., which groups of nodes are (un)likely to be controlled by the same entity).
- To which nodes do *dependencies* exist (e.g., for business reasons).

Adding nodes of organizations one interacts with to one’s quorum sets might be necessary to maintain “sync” with these organizations [13], as opposed to ending up with diverging ledgers in the event of a fork.

In the following discussion, we will use *graph representations* for modeling individual preferences. It is an intriguing hypothesis that the FBAS paradigm can enable Sybil-resistant and yet energy-efficient permissionless consensus by bootstrapping quorum systems along existing trust graphs or interdependence graphs. In Sec. 3.1 we saw that transforming an FBAS into an equally sized regular graph leads to a loss of information, i.e., can yield only heuristic representations. In the following sections we pose the inverse question: How can a “good” FBAS (\mathbf{V}, \mathbf{Q}) be instantiated from a given graph $G = (\mathbf{V}, E)$?

For evaluating example policies incorporating individual preferences, we will use the *autonomous system* (AS) relationships graph inferred by the CAIDA project⁹—a reflection of the interdependence and trust between networks that form the Internet. The topological structure of the Internet has repeatedly been cited as an argument for the viability of the FBAS model [13, 16]. We discuss results based on two snapshots of the AS relations graph: from January 1998—the earliest available snapshot describing a younger Internet with 3233 ASs connected via 4921 (directed) customer/provider links and 852 (undirected) peering links—and from January 2020—with 67308 ASs connected via 133864 customer/provider links and 312763 peering links. We will refer to the graphs as G_{AS98} and G_{AS20} .

6.2 Naive individualistic QSC

We consider a QSC policy naively individualistic if it is based entirely on *individual preferences*. We model “preference for a node” as edges in a graph $G = (\mathbf{V}, E)$, with nodes being aware only of their own graph neighborhood.

Consider a simple representative of this class—forming quorum sets using the entire graph neighborhood of a node, weighing each neighbor equally within a $3f + 1$ threshold logic (that models the assumption that strictly less than a third of all neighbors can be faulty):

$$\forall v \in \mathbf{V} : U = \{v\} \cup \{v' \in \mathbf{V} \mid (v, v') \in E\}$$

$$\mathbf{Q}(v) = \text{qset}(v, (U, \emptyset, \lceil \frac{2|U| + 1}{3} \rceil))$$

(All Neighbors QSC)

If G is a complete graph, we get the same result as with **Ideal Open QSC**. If G is not connected, we cannot have quorum intersection (and hence safety). The latter is also true if G contains more than one cluster of sufficient size and weak (relative) connectedness to the rest of the graph. We can confirm that this is the case for the AS graph snapshots G_{AS98} and G_{AS20} . Using them, **All Neighbors QSC** induces FBASs

that *do not* enjoy quorum intersection¹⁰. The high prevalence of AS peering is a likely explanation for why sufficiently well intraconnected clusters can emerge outside of the “natural” top tier of the AS graph.

A lack of quorum intersection implies that the induced FBASs may split into multiple sub-FBASs. This might be a desirable effect when bootstrapping from individual preferences. For example, separated communities with low levels of inter-community interaction and trust might prefer the added sovereignty of an “own” FBAS. We repeated the analysis for the respectively largest sub-FBASs, with an upper bound on top tier size¹¹ of, respectively, 355 and 14339 nodes. Potential top tier sizes of this magnitude make a complete analysis unfeasible (s.a. the discussion on analysis scalability in Sec. 5.6). This is problematic, as the robustness of the resulting FBASs, in terms of safety and liveness, cannot be reliably determined. Existing weaknesses in the global quorum structure cannot be identified and (strategically) fixed. Weaknesses, however, are likely to exist. For example, preliminary analysis results for the FBAS instantiated from G_{AS98} imply the existence of blocking sets with only 3 members.

6.3 Tier-based QSC

Towards making resulting top tiers more focused (and hence, the resulting FBASs more efficient and more amenable to analysis), QSC policies can incorporate *strategic considerations* in addition to individual preferences. We explore a prudent example strategy in the following: the weighing of nodes based on *tierness*, or relative importance. Tierness is an established notion for ASs in the Internet graph. For FBASs, a tiered quorum structure with every node including only higher-tier neighbors in its quorum sets was proposed (as an example) as early as in the original FBAS proposal [16]. Classifying nodes based on their tierness is also related to the *quality*-based configuration format currently used by the Stellar software [13]. Lastly, it is a plausible assumption that the relative tierness of graph neighbors can be estimated locally, enabling QSC decisions that do not require a global view.

We sketch an example QSC policy in which nodes use only higher-tier nodes in their quorum sets, or same-tier nodes if none of their neighbor appears to be of higher tier. We assume that nodes can infer the relative tierness of their graph neighbors. Specifically, that they can determine which of their neighbors are of a higher tier than themselves. For simulation, we use the PageRank [19] score of nodes (calculated without dampening) as a proxy for their tier-

¹⁰ As determined using `fbas_analyzer` (Sec. 5).

¹¹ Based on the size of the largest quorum that is fully contained in a strongly connected component (which is the union of all such quorums).

⁹ The CAIDA AS Relationships Dataset, 1998-01-01 (serial-1) and 2020-01-01 (serial-2), <https://www.caida.org/data/as-relationships/>

ness. Each simulated node considers a neighbor of higher (lower) tier if the neighbor’s PageRank score is twice as high (low) as its own. More formally, with $R(v)$ denoting the PageRank score of node v , $\text{edges}^+(v)$ the set of its neighbors ($\text{edges}^+(v) := \{v' \in \mathbf{V} \mid (v, v') \in E\}$), H its higher-tier neighbors and P its same-tier neighbors (“peers”):

$$\begin{aligned}
 H(v) &= \{v' \in \text{edges}^+(v) \mid R(v') \geq 2R(v)\} \\
 P(v) &= \{v' \in \text{edges}^+(v) \mid \frac{1}{2}R(v) < R(v') < 2R(v)\}
 \end{aligned}$$

(Tierness Heuristics)

Based on this heuristic, we can define the following QSC policy:

$$\forall v \in \mathbf{V} : U = \begin{cases} \{v\} \cup H(v) & \text{if } H(v) \neq \emptyset \\ \{v\} \cup P(v) & \text{else} \end{cases}$$

$$\mathbf{Q}(v) = \text{qset}(v, (U, \emptyset, \lceil \frac{2|U| + 1}{3} \rceil))$$

(Higher-Tier Neighbors QSC)

Our results show that improvements to the naive case are possible when incorporating strategic considerations, despite the fact that the quorum structure is heavily influenced by individual preferences. More prominently—top tiers become of more manageable size (both for analysis and for consensus protocols leveraging the FBAS).

We simulated the application of **Higher-Tier Neighbors QSC** using the AS graph snapshots G_{AS98} and G_{AS20} . The two thus induced FBASs contained, respectively, 2 and 6 nodes with one-node quorums sets which we filter out for the subsequent analysis. We apply `fbas_analyzer`, our software-based analysis framework (cf. Sec. 5), to the resulting FBASs.

Figure 4 presents the analysis findings. It depicts histograms of the relevant sets, i.e., how many minimal quorums, minimal blocking sets or minimal splitting sets of a given size exist for the given FBAS. For the G_{AS98} case, we restricted our minimal splitting sets analysis to the core of the FBAS, i.e., to its top tier and all nodes that are referenced by top tier nodes either directly or transitively¹². We find that doing so yields more informative results; the full FBAS contains a large number of splitting sets with cardinality 1 that only split off very small groups of nodes from the rest. Even when restricting the analysis to core nodes only, we were not able to fully enumerate the minimal splitting sets for G_{AS20} in reasonable time, due to the size and specific structure of the resulting FBAS.

Strikingly, our analysis reveals that the liveness of both FBASs is easily compromised. Despite their relatively large

¹² This corresponds to the union of all strongly connected components that contain a quorum.

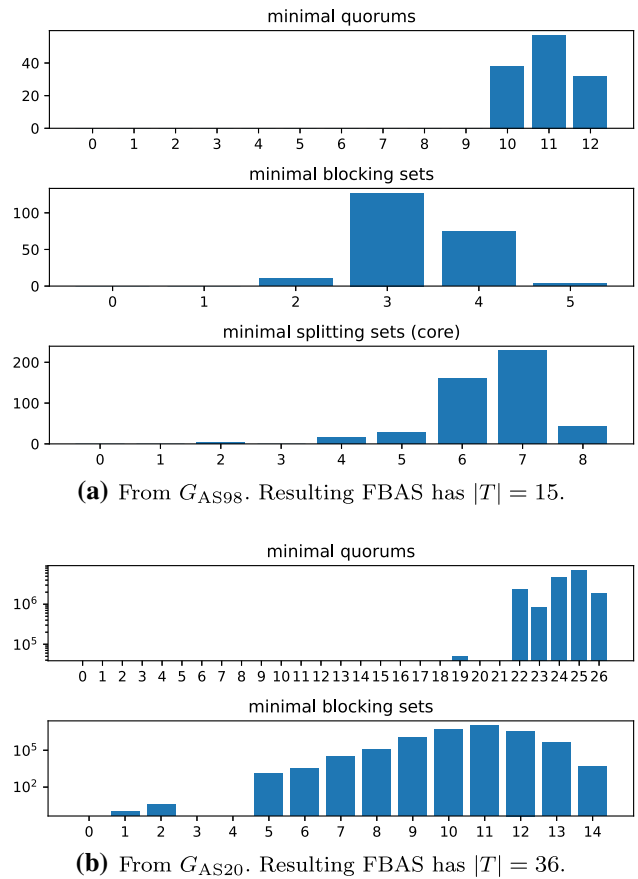


Fig. 4 Histogram of the cardinalities of relevant sets in FBASs resulting from the application of **Higher-Tier Neighbors QSC** using snapshots of the AS relationship graph (G_{AS98} , G_{AS20})

top tiers (of 15 and 36 nodes, respectively), groups of only 2 nodes, and in the G_{AS20} case even one group of only one node, exist that are sufficient to completely block (or censor) the FBAS. For comparison, symmetric top tiers of the same size would result in all minimal blocking sets having sizes of, respectively, 5 and 12. This liveness-threatening discrepancy can be explained through *cascading failures*: If (for example) two nodes fail, this can result in a third node with a “weak” quorum set becoming unsatisfiable, so that three nodes have now de-facto failed, which can result in a fourth node becoming unsatisfiable, et cetera. It can be concluded that the composition and size of smallest blocking sets for an FBAS is heavily influenced by the “weakest” quorum sets in the FBAS’ top tier. An additional example for cascading failures is given Appendix B.

6.4 Symmetry enforcement

The graph-based QSC policies discussed so far easily result in systems that are brittle (in the sense of small minimal blocking sets) and hard to analyze. Both of these characteristics are vastly improved, relative to top tier size, in FBASs

with symmetric top tiers. However, symmetric top tiers emerge organically from a preexisting relationship graph G only if the top tier nodes form a complete subgraph of G , which is not the case in the graphs investigated so far. As a policy enhancement, nodes believing themselves to be top tier can *mirror* the quorum sets of other apparently top tier nodes, strategically including non-neighbors in their quorum sets for improving the global FBAS structure. A behavior along this lines can, in fact, be observed in the live Stellar network (s.a. Appendix C).

Yet, by making validator decisions independent of the local knowledge representation G , new assumptions become necessary to be able to rule out attacks. Mirroring makes it easier for malicious top tier nodes to introduce Sybil nodes into the top tier. The approach is therefore only secure (w.r.t. both safety and liveness) if it can be assumed that nodes in T make plausibility checks before expanding their quorum sets, so that attempted (Sybil) attacks can be detected. Given the lack of explicit incentives for running validator nodes in systems like Stellar, such a burden on the operators of top tier nodes might be viewed as problematic [11]. However, similar critique can also be voiced against systems (like Bitcoin) that base their security arguments on notions of economic rationality, as economic rationality can also be leveraged by attackers [6].

7 Limits on openness and top tier fluidity

The FBAS paradigm reportedly enables the instantiation of consensus systems with *open membership* [13,16]. And clearly, arbitrary nodes can join an FBAS, causing new quorums to be formed that contain them. Based on the preceding discussion, however, we recognize that without creating a new, de-facto disjoint FBAS, or the active reconfiguration of existing nodes, new nodes cannot become part of *minimal* quorums and hence minimal blocking sets. Thereby, their existence is irrelevant as far as the discussed liveness indicators are concerned, and their importance for safety is limited. In Sec. 4 we defined the notion of a *top tier* to reflect the set of nodes in an FBAS that is central to liveness, i.e., the set of nodes from which all minimal quorums and blocking sets are formed. The top tier wields absolute power to censor and block the whole FBAS.

In the following, we investigate the question to what extent this top tier can be considered a group with open membership. How can its power be diluted by promoting additional nodes to top tier status? Can nodes be “fired” from the top tier? We make the case that, in general, a top tier T can neither grow nor shrink without either the active involvement of existing top tier nodes *or* a loss of safety guarantees. We base all subsequent projections on the status quo of an FBAS that

enjoys quorum intersection despite faulty nodes (a *safe* FBAS as per the discussion in Sec. 3.4).

7.1 Top-down top tier change

As a preliminary remark, recall that, as per Def. 4.4, we define the top tier T of an FBAS (\mathbf{V}, \mathbf{Q}) as the union of all its minimal quorums. T is therefore also a quorum and intersects every quorum in (\mathbf{V}, \mathbf{Q}) .

Theorem 7.1 (top tier can safely change itself) *Let $T \subset \mathbf{V}$ be the top tier of an FBAS (\mathbf{V}, \mathbf{Q}) that enjoys quorum availability and quorum intersection. Then it is possible, without compromising neither quorum availability nor quorum intersection, to instantiate a new top tier $T' \subseteq \mathbf{V}$, $T' \neq \emptyset$ by changing only the quorum sets of new and old top tier nodes $v \in T \cup T'$.*

Proof Let $T' \subseteq \mathbf{V}$, $T' \neq \emptyset$ be the target top tier. Let \mathbf{Q}' be a modification of \mathbf{Q} so that $\forall v \in T \cup T' : \mathbf{Q}'(v) = \{T'\}$ ¹³ and $\forall v \notin T \cup T' : \mathbf{Q}'(v) = \mathbf{Q}(v)$. As T' is a quorum w.r.t. \mathbf{Q}' , (T', \mathbf{Q}') enjoys quorum availability. Therefore, $(\mathbf{V}, \mathbf{Q}')$ enjoys quorum availability. $(\mathbf{V} \setminus T', \mathbf{Q}')$ does not enjoy quorum availability, because no node in T is satisfied without T' and no node in $\mathbf{V} \setminus T$ can form a quorum without a node from T (otherwise T would not have been the top tier w.r.t. \mathbf{Q} , cf. Def. 4.4). There are therefore no quorums w.r.t. \mathbf{Q}' that are disjoint of T' . $(\mathbf{V}, \mathbf{Q}')$ therefore enjoys quorum intersection iff (T', \mathbf{Q}') enjoys quorum intersection, which it (trivially) does. \square

The situation is less clear if some nodes $T \setminus T'$ do not wish to leave T . Note, however, that single nodes can always endanger safety via trivial configurations such as $\mathbf{Q}(v) = \{\{v\}\}$. If performed by one or more nodes in T , such an act of sabotage can have an impact on the safety of large portions of the FBAS.

7.2 Bottom-up top tier change

In the following, we assume a “self-centered” top tier in the sense that all top tier nodes include only other top tier nodes in quorum sets. Symmetric top tiers (Def. 4.5) have this property, as do top tiers observed in the wild in the Stellar network (cf. Appendix C).

Theorem 7.2 (no safe top tier change with uncooperative top tier) *Let (\mathbf{V}, \mathbf{Q}) be an FBAS that enjoys quorum intersection and has a “self-centered” top tier $T \subset \mathbf{V}$ such that all top tier quorum slices are comprised of only top tier nodes ($\forall v \in \mathbf{V} : \bigcup \mathbf{Q}(v) \subseteq T$). Then it is **not** possible, without*

¹³ Without loss of generality. Clearly, more robust top tier constructions are possible.

compromising quorum intersection, to instantiate a new top tier $T' \subseteq \mathbf{V}$, $T' \neq T$ by changing only the quorum sets of non-top tier nodes $v \in \mathbf{V} \setminus T$.

Proof Let $T' \subseteq \mathbf{V}$, $T' \neq T$ be the top tier of a new FBAS $(\mathbf{V}, \mathbf{Q}')$ that enjoys quorum intersection. Let $\hat{\mathcal{U}}$ and $\hat{\mathcal{U}}'$ be the sets of all minimal quorums of (\mathbf{V}, \mathbf{Q}) and $(\mathbf{V}, \mathbf{Q}')$, respectively. As per Def. 4.4, $T' \neq T$ implies that $\hat{\mathcal{U}} \neq \hat{\mathcal{U}}'$.

Assume there exists a $\hat{U} \in \hat{\mathcal{U}} \setminus \hat{\mathcal{U}}'$. Then \hat{U} is a quorum w.r.t. \mathbf{Q} and either (a) not a quorum w.r.t. \mathbf{Q}' or (b) not minimal w.r.t. \mathbf{Q}' .

However, we require that the quorum sets of top tier nodes don't change: $\forall v \in T : \mathbf{Q}'(v) = \mathbf{Q}(v)$. Therefore \hat{U} is a quorum also w.r.t. \mathbf{Q}' , contradicting (a). Hence, (b) must hold and there must be a $\hat{U}' \in \hat{\mathcal{U}}'$ such that $\hat{U}' \subset \hat{U}$ (cf. Def. 4.1). As $\hat{U}' \subseteq \hat{U} \subseteq T$, \hat{U}' being a quorum w.r.t. \mathbf{Q}' implies it also being a quorum w.r.t. \mathbf{Q} . But then \hat{U} is not minimal w.r.t. \mathbf{Q} , implying $\hat{U} \notin \hat{\mathcal{U}}$ and thus again leading to a contradiction. This proves that $\hat{\mathcal{U}} \subseteq \hat{\mathcal{U}}'$.

Assume now there exists a $\hat{U}' \in \hat{\mathcal{U}}' \setminus \hat{\mathcal{U}}$ and let $\hat{U} \in \hat{\mathcal{U}}$. As $(\mathbf{V}, \mathbf{Q}')$ enjoys quorum intersection, $\hat{U}' \cap \hat{U} \neq \emptyset$ and \hat{U}' contains members of the “old” top tier T . \hat{U}' is a quorum w.r.t. \mathbf{Q}' , but $\hat{U}' \cap T$ cannot be a quorum w.r.t. \mathbf{Q}' as otherwise \hat{U}' would not be a minimal quorum. There must therefore exist a node $v \in \hat{U}' \cap T$ with a quorum slice $q \in \mathbf{Q}'(v)$ such that $(\hat{U}' \cap T) \subset q \subseteq \hat{U}'$ (cf. Def. 3.4), i.e., $q \setminus T \neq \emptyset$. As $v \in T$, we require that $\mathbf{Q}'(v) = \mathbf{Q}(v)$ and $\bigcup \mathbf{Q}(v) \subseteq T$, which leads to a contradiction since $q \in \mathbf{Q}(v)$ and $q \setminus T \neq \emptyset$. It must therefore hold that $\hat{\mathcal{U}} \setminus \hat{\mathcal{U}}' = \emptyset$, $\hat{\mathcal{U}} = \hat{\mathcal{U}}'$ and $T = T'$. \square

7.3 Consequences

Who determines which FBAS nodes get to form the top tier? Our results imply that, if maintaining safety is seen as an untouchable requirement, the top tier T_i of an FBAS $(\mathbf{V}_i, \mathbf{Q}_i)$ at “iteration” i is legitimated by decisions of, exclusively, members of $T_{i-1} \cup T_i$ (if none of them cooperates, we lose safety, if all of them cooperate, we don't). Because of the top tier's importance to the liveness, safety and performance achievable within a given FBAS, open membership in \mathbf{V}_i is of little benefit without open membership in T_i .

How closed is the membership in T_i ? It might be sufficient that only some nodes in T_{i-1} support a transition to T_i . If reactive QSC policies are used (e.g., for enforcing top tier symmetry as discussed in Sec. 6.4), one cooperative top tier node $v \in T_{i-1}$ might already be enough for growing the top tier in a way that is robust and doesn't only dilute the relative influence of v . How partially supported top tier changes would play out must be investigated based on more specific scenarios. We expect the safe “firing” of top tier nodes to be especially challenging.

Which begs the question—can the safety requirement be weakened? For example, given sufficiently good (out-

of-band) coordination between members of $\mathbf{V}_{i-1} \setminus T_{i-1}$, a $(\mathbf{V}_i, \mathbf{Q}_i)$ might be instantiated in which at least $(\mathbf{V}_i \setminus T_{i-1}, \mathbf{Q}_i)$ enjoys quorum intersection. It is conceivable that novel protocols can be developed, possibly also leveraging the FBAS structure, that reduce the notorious difficulty of coordinating such bottom-up actions.

8 Conclusion

We demonstrate in this paper that, despite the complexity of the FBAS model, the properties of concrete FBAS instances can be described in a way that is both precise and intuitive, and allows comparisons with more classical Byzantine agreement systems. We propose the notions of minimal blocking sets, minimal splitting sets and top tiers to describe which groups of nodes can compromise liveness and safety. In essence, minimal blocking sets and minimal splitting sets describe minimal viable threat scenarios, thereby enabling a comprehensive risk assessment in FBAS-based systems like the Stellar network. While some analyses imply computational problems of exponential complexity, we developed and implemented algorithms that enable the exact analysis of a wide range of interesting FBASs.

Our implemented analysis framework also enables us to investigate how individual configurations result in global properties. We find that overly strategic configuration policies result in FBASs that are indistinguishable from permissioned systems. Individualistic approaches, on the other hand, cannot guarantee safe results while quickly resulting in systems that are infeasible to analyze. Adding some strategic decision-making at organically emerging top tier nodes offers a potential middle way towards robust FBASs instantiated from the sum of individual preferences.

Independently of the way in which a given FBAS came to be, however, the composition of a once established top tier cannot be influenced without the cooperation of existing top tier nodes, without at the same time threatening safety. This seems to place the FBAS paradigm closer to the “permissioned consensus” camp than hoped. More investigation is needed to determine the exact impact of bottom-up top tier changes (as in number of nodes affected by a loss of safety or liveness, for example) and to formulate possible coordination strategies to keep such impacts low.

Acknowledgements We thank Ben Schumacher, Jakob Hoffmann and pieterjan84 for helpful discussions. We thank Ingolf Pernice, Rainer Böhme and Patrik Keller for providing valuable feedback at various stages of this work. We thank the anonymous reviewers of this work for their insightful comments and suggestions.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was funded by the German Federal Ministry of Education and Research (BMBF) through its funding for the Weizenbaum Institute for the Networked Society.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Additional corollaries, theorems and proofs

A.1 Minimal quorums

Corollary A.1 (minimal quorum intersection \iff quorum intersection) *Let $\mathcal{U} \subseteq 2^V$ be the set of all quorums of the FBAS (\mathbf{V}, \mathbf{Q}) , $\hat{\mathcal{U}} \subseteq \mathcal{U}$ be the set of all minimal quorums. All pairs of $U_1, U_2 \in \mathcal{U}$ intersect iff all pairs of $\hat{U}_1, \hat{U}_2 \in \hat{\mathcal{U}}$ intersect.*

Proof Since $\hat{\mathcal{U}} \subseteq \mathcal{U}$, $\forall U_1, U_2 \in \mathcal{U} : U_1 \cap U_2 \neq \emptyset$ trivially implies that $\forall \hat{U}_1, \hat{U}_2 \in \hat{\mathcal{U}} : \hat{U}_1 \cap \hat{U}_2 \neq \emptyset$. The other direction follows because $\forall U_1, U_2 \in \mathcal{U} \exists \hat{U}_1, \hat{U}_2 \in \hat{\mathcal{U}} : \hat{U}_1 \subseteq U_1 \wedge \hat{U}_2 \subseteq U_2$ ($\hat{\mathcal{U}}$ being the set of all minimal sets w.r.t. \mathcal{U} ; s.a. Def. 4.1). If all pairs in $\hat{\mathcal{U}}$ intersect, so must therefore all pairs in \mathcal{U} . \square

This was previously also shown in [12].

A.2 Blocking sets

Corollary A.2 (blocking for all \implies blocking for all minimal) *Let $\mathcal{U} \subseteq 2^V$ be the set of all quorums of the FBAS (\mathbf{V}, \mathbf{Q}) , and $\hat{\mathcal{U}} \subseteq \mathcal{U}$ be the set of all minimal quorums. If B is a blocking set for \mathcal{U} , then it is also a blocking set for $\hat{\mathcal{U}}$.*

Proof B is a blocking set for $\mathcal{U} \iff \forall U \in \mathcal{U} : B \cap U \neq \emptyset$ (Def. 4.2). $\hat{\mathcal{U}} \subseteq \mathcal{U} \implies \forall \hat{U} \in \hat{\mathcal{U}} : B \cap \hat{U} \neq \emptyset$, so that B is also a blocking set for $\hat{\mathcal{U}}$. \square

Corollary A.3 (blocking for all minimal \implies blocking for all) *Let $\mathcal{U} \subseteq 2^V$ be the set of all quorums of the FBAS (\mathbf{V}, \mathbf{Q}) , and $\hat{\mathcal{U}} \subseteq \mathcal{U}$ be the set of all minimal quorums. If B is blocking set for $\hat{\mathcal{U}}$, then it is also a blocking set for \mathcal{U} .*

Proof B is a blocking set for $\hat{\mathcal{U}} \implies \forall U \in \hat{\mathcal{U}} : B \cap U \neq \emptyset$ (Def. 4.2). $\hat{\mathcal{U}} \subseteq \mathcal{U}$ and all $U \in \hat{\mathcal{U}}$ are minimal w.r.t. $\mathcal{U} \implies \forall U \in \mathcal{U} \exists \hat{U} \in \hat{\mathcal{U}} : \hat{U} \subseteq U$ (cf. Def. 4.1) $\implies U \cap B \neq \emptyset \implies B$ is blocking for all $U \in \mathcal{U}$. \square

Corollary A.4 (minimal blocking sets result from minimal quorums) *Let $\mathcal{U} \subseteq 2^V$ be the set of all quorums of the FBAS*

(\mathbf{V}, \mathbf{Q}) , $\hat{\mathcal{U}} \subseteq \mathcal{U}$ be the set of all minimal quorums, and $\hat{\mathcal{B}} \subseteq 2^V$ be the set of all minimal blocking sets. Then each minimal blocking set $\hat{B} \in \hat{\mathcal{B}}$ of the FBAS is minimally blocking w.r.t. $\hat{\mathcal{U}}$, i.e., \hat{B} intersects every minimal quorum $\hat{U} \in \hat{\mathcal{U}}$ and no $B' \subset \hat{B}$ intersects every minimal quorum $\hat{U} \in \hat{\mathcal{U}}$.

Proof Let $\mathcal{B} \subseteq 2^V$ be the set of all blocking sets w.r.t. $\hat{\mathcal{U}}$. Based on Cor. A.2 and Cor. A.3, \mathcal{B} is exactly the set of all blocking sets for \mathcal{U} . Hence the set of all minimal sets w.r.t. \mathcal{B} is exactly the set of all minimal blocking sets w.r.t. \mathcal{U} and therefore the set of all minimal blocking sets for (\mathbf{V}, \mathbf{Q}) , or $\hat{\mathcal{B}} \subseteq \mathcal{B}$. Likewise, as \mathcal{B} is the set of all blocking sets w.r.t. $\hat{\mathcal{U}}$, $\hat{\mathcal{B}}$ is the set of all minimal blocking sets w.r.t. $\hat{\mathcal{U}}$. \square

A.3 Splitting sets

Definition A.1 (quorum expanders) For an FBAS (\mathbf{V}, \mathbf{Q}) , a *quorum expander* is any node $v \in \mathbf{V}$ that is part of a quorum slice $q \in \mathbf{Q}(v')$ of another node $v' \in \mathbf{V}$ that is not a quorum slice for v , i.e., any node $v \in \mathbf{V}$ for which $\exists v' \in \mathbf{V}, q' \in \mathbf{Q}(v') : v \in q' \wedge (\forall q \in \mathbf{Q}(v) : q \not\subseteq q')$.

Theorem A.1 (minimal splitting sets formed exclusively of quorum expanders and top tier nodes) *Let $\hat{\mathcal{S}} \subseteq 2^V$ be the set of all minimal splitting sets of the FBAS (\mathbf{V}, \mathbf{Q}) , $X \subseteq \mathbf{V}$ the set of all quorum expanders of the FBAS (Def. A.1) and $T \subseteq \mathbf{V}$ the top tier of the FBAS (the union of all minimal quorums, Def. 4.4). Then it holds that $\bigcup \hat{\mathcal{S}} \subseteq T \cup X$.*

Proof Let $\hat{S} \in \hat{\mathcal{S}}$ and $s \in \hat{S}$ be an arbitrary node in that splitting set. We show that $s \in T$ or $s \in X$ must hold.

\hat{S} is a minimal splitting set, therefore $\hat{S} \setminus \{s\}$ is not a splitting set for any s . Consequently, $(\mathbf{V}, \mathbf{Q})^{\hat{S} \setminus \{s\}}$ enjoys quorum intersection while $(\mathbf{V}, \mathbf{Q})^{\hat{S}}$ doesn't. Let $\hat{U}_1, \hat{U}_2 \subseteq \mathbf{V}, \hat{U}_1 \cap \hat{U}_2 = \emptyset$ be two non-intersecting minimal quorums in $(\mathbf{V}, \mathbf{Q})^{\hat{S}}$ such that \hat{U}_1 does not contain a quorum in $(\mathbf{V}, \mathbf{Q})^{\hat{S} \setminus \{s\}}$. (If both \hat{U}_1 and \hat{U}_2 contained quorums in $(\mathbf{V}, \mathbf{Q})^{\hat{S} \setminus \{s\}}$, the FBAS would lack quorum intersection.)

If $\hat{U}_1 \cup \{s\}$ contains a quorum in $(\mathbf{V}, \mathbf{Q})^{\hat{S} \setminus \{s\}}$, then $\hat{U}_1 \cup \{s\}$ contains a minimal quorum $\hat{U}'_1 \subseteq \hat{U}_1 \cup \{s\}$ that contains s . Consequently, s is part of the top tier T' of $(\mathbf{V}, \mathbf{Q})^{\hat{S} \setminus \{s\}}$, i.e., $s \in T'$. As the only effect of the delete operation (Def. 3.7) on \mathbf{Q} is to remove nodes from quorum slices and both (\mathbf{V}, \mathbf{Q}) and $(\mathbf{V}, \mathbf{Q})^{\hat{S} \setminus \{s\}}$ enjoy quorum intersection, it holds that $T' \subseteq T$ (the proof is analogous to the proof of Thm. 7.2). Consequently, $s \in T$.

If $\hat{U}_1 \cup \{s\}$ does not contain a quorum in $(\mathbf{V}, \mathbf{Q})^{\hat{S} \setminus \{s\}}$, then, because \hat{U}_1 is a quorum in $(\mathbf{V}, \mathbf{Q})^{\hat{S}}$, the forming of a quorum fails because of s . For $(\mathbf{V}', \mathbf{Q}') := (\mathbf{V}, \mathbf{Q})^{\hat{S} \setminus \{s\}}$, it must hold that $\exists v \in \hat{U}_1, \exists q \in \mathbf{Q}'(v) : q \subseteq \hat{U}_1 \cup \{s\}$ while $\forall q' \in \mathbf{Q}'(s) : q \not\subseteq \hat{U}_1 \cup \{s\}$. The node s is therefore one of the quorum expanders X' of $(\mathbf{V}, \mathbf{Q})^{\hat{S} \setminus \{s\}}$, i.e., $s \in X'$. It trivially holds that $X' \subseteq X$ and, therefore, $s \in X$. \square

A.4 Top tier

Corollary A.5 (minimal blocking sets formed exclusively of top tier nodes) *Let T be the top tier of an FBAS (\mathbf{V}, \mathbf{Q}) , and $\hat{\mathcal{B}} \subseteq 2^{\mathbf{V}}$ be the set of all minimal blocking sets of (\mathbf{V}, \mathbf{Q}) . Then $\forall \hat{B} \in \hat{\mathcal{B}} : \hat{B} \subseteq T$.*

Proof From Cor. A.4 it follows that all $\hat{B} \in \hat{\mathcal{B}}$ are formed of nodes contained in at least one minimal quorum $\hat{U} \in \hat{\mathcal{U}}$. As $T = \bigcup \hat{U}$ (Def. 4.4), $\forall \hat{B} \in \hat{\mathcal{B}} : \hat{B} \subseteq T$. \square

Theorem A.2 (each top tier node in at least one minimal blocking set) *Let T be the top tier of an FBAS (\mathbf{V}, \mathbf{Q}) , and $\hat{\mathcal{B}} \subseteq 2^{\mathbf{V}}$ be the set of all minimal blocking sets of (\mathbf{V}, \mathbf{Q}) . Then for each top tier node $v \in T$ there is at least one minimal blocking set $\hat{B} \in \hat{\mathcal{B}}$ such that $v \in \hat{B}$.*

Proof Let $v \in T$ be an arbitrary top tier node and $\hat{U} \in \hat{\mathcal{U}}$ an arbitrary minimal quorum such that $v \in \hat{U}$ (recall that $T = \bigcup \hat{U}$; Def. 4.4). $T \setminus \hat{U}$ intersects every $\hat{U}' \in \hat{\mathcal{U}} \setminus \{\hat{U}\}$, as otherwise there would be a $\hat{U}' \in \hat{\mathcal{U}}$ such that $\hat{U}' \subset \hat{U}$ (i.e., \hat{U} would not be a minimal quorum). Therefore, $T \setminus \hat{U}$ is a blocking set w.r.t. $\hat{\mathcal{U}} \setminus \{\hat{U}\}$ and $B' = \{v\} \cup T \setminus \hat{U}$ is a blocking set w.r.t. $\hat{\mathcal{U}}$. $B' \setminus \{v\}$ is not a blocking set w.r.t. $\hat{\mathcal{U}}$ because it doesn't intersect \hat{U} . Hence, all $\hat{B} \in \hat{\mathcal{B}}$ such that $\hat{B} \subseteq B'$ (and there must be at least one— B' —because B' is a blocking set w.r.t. $\hat{\mathcal{U}}$) must contain v . Hence the FBAS has at least one $\hat{B} \in \hat{\mathcal{B}}$ that contains v . \square

Theorem A.3 (Blocking sets in non-nested symmetric top tier) *For an FBAS (\mathbf{V}, \mathbf{Q}) with a symmetric top tier $T \subseteq \mathbf{V}$, $m := |T|$ such that $\forall v \in T : \mathbf{Q}(v) = \text{qset}(v, (T, \emptyset, t))$ it holds that: All minimal blocking sets $\hat{B} \in \hat{\mathcal{B}}$ have cardinality $\max(m - t + 1, 0)$.*

Proof We observe that for any $v \in T$, $\mathbf{Q}(v) = \{q \subseteq \mathbf{V} : v \in q\} \wedge |q \cap T| \geq t$ (Def. 3.2 and 3.3). A $U \subset T$ is therefore a quorum in (\mathbf{V}, \mathbf{Q}) iff $|U| \geq t$ (Def. 3.4). As all $U \subset T$ with $|U| \geq t$ are quorums in (\mathbf{V}, \mathbf{Q}) , the minimal quorums in (\mathbf{V}, \mathbf{Q}) are exactly $\hat{\mathcal{U}} = \{\hat{U} \subseteq T, |\hat{U}| = t\}$. Then:

For all $B \subseteq T$ with $|B| = m - t + 1$ it holds that $\forall U' \subseteq T \setminus B : |U'| = t - 1 < t$. Hence, no $U' \subseteq T \setminus B$ is a quorum, there are no quorums that are disjoint with B and B is a blocking set (Def. 4.2). B is furthermore a minimal blocking set, as for any $B' \subset B$ it holds that $U = T \setminus B'$ is a quorum (as $|U| \geq t$), and so B' is not a blocking set. \square

Theorem A.4 (Splitting sets in non-nested symmetric top tier) *For an FBAS (\mathbf{V}, \mathbf{Q}) that consists entirely of a symmetric top tier $T = \mathbf{V}$, $m := |T|$ such that $\forall v \in \mathbf{V} : \mathbf{Q}(v) = \text{qset}(v, (\mathbf{V}, \emptyset, t))$ it holds that all minimal splitting sets $\hat{S} \in \hat{\mathcal{S}}$ have cardinality $\max(2t - m, 0)$.*

Proof Like in Thm. A.3, we observe that the minimal quorums in (\mathbf{V}, \mathbf{Q}) are exactly $\hat{\mathcal{U}} = \{\hat{U} \subseteq T, |\hat{U}| = t\}$. Then:

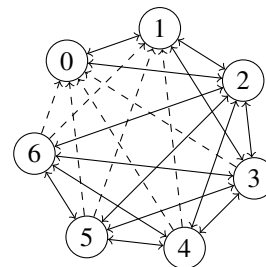
Let $\hat{S} \in \hat{\mathcal{S}}$ be an arbitrary minimal splitting set for (\mathbf{V}, \mathbf{Q}) . If $2t - m \leq 0$, there exist two minimal quorums $\hat{U}_1, \hat{U}_2 \in \hat{\mathcal{U}}$ (with cardinality t) that do not intersect. There is then only one $\hat{S} = \emptyset$ and the cardinality of all minimal splitting sets is trivially 0. In the following, we assume that $2t - m > 0$ and (\mathbf{V}, \mathbf{Q}) therefore enjoys quorum intersection. Since (\mathbf{V}, \mathbf{Q}) consists entirely of a symmetric top tier, no $v \in \mathbf{V}$ is a quorum expander. Splitting sets must therefore contain an intersection of at least one pair of minimal quorums (for illustration, cf. the proof of Thm. A.1). There are therefore at least two minimal quorums $\hat{U}_1, \hat{U}_2 \in \hat{\mathcal{U}}$ such that $\hat{S} = \hat{U}_1 \cap \hat{U}_2$. Let $U = \hat{U}_1 \cup \hat{U}_2$. $N' = T \setminus U$ must be empty, otherwise we could, with an arbitrary $N'' \subseteq \hat{S}$, $|N''| = |N'|$ find a minimal quorum $\hat{U}_3 = (\hat{U}_2 \setminus N'') \cup N'$ such that $\hat{U}_1 \cap \hat{U}_3 \subset \hat{S}$ (i.e., \hat{S} is not minimal). It therefore holds that $U = T$ and, since, $|\hat{U}_1| = |\hat{U}_2| = t$, $|\hat{S}| = 2t - m$. \square

B Example analysis: Toy network with cascading failures

Consider the FBAS (\mathbf{V}, \mathbf{Q}) with $\mathbf{V} = \{0, 1, 2, 3, 4, 5, 6\}$ and \mathbf{Q} such that:

- $\mathbf{Q}(0) = \text{qset}(0, (\{0, 1, 2\}, \emptyset, 3))$
- $\mathbf{Q}(1) = \text{qset}(1, (\{0, 1, 2, 3\}, \emptyset, 3))$
- $\mathbf{Q}(2) = \text{qset}(2, (\{0, 1, 2, 3, 4, 5, 6\}, \emptyset, 5))$
- $\mathbf{Q}(3) = \text{qset}(3, (\{0, 1, 2, 3, 4, 5, 6\}, \emptyset, 5))$
- $\mathbf{Q}(4) = \text{qset}(4, (\{0, 1, 2, 3, 4, 5, 6\}, \emptyset, 5))$
- $\mathbf{Q}(5) = \text{qset}(5, (\{0, 1, 2, 3, 4, 5, 6\}, \emptyset, 5))$
- $\mathbf{Q}(6) = \text{qset}(6, (\{0, 1, 2, 3, 4, 5, 6\}, \emptyset, 5))$

This \mathbf{Q} can be the result of a scenario in which all $v \in \mathbf{V}$ apply the QSC policy **All Neighbors QSC** (Sec. 6.2) based on following graph G (unidirectional edges highlighted as dashed lines):



We find the minimal blocking sets $\hat{\mathcal{B}} \subset 2^{\mathbf{V}}$ of (\mathbf{V}, \mathbf{Q}) using our analysis tool (cf. Sec. 5):

$$\hat{\mathcal{B}} = \{\{2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{1, 6\}, \{0, 3\}, \{3, 4, 5\}, \{3, 4, 6\}, \{3, 5, 6\}, \{0, 4, 5\}, \{0, 4, 6\}, \{0, 5, 6\},$$

{4, 5, 6}}

Despite the fact that most nodes in \mathbf{V} have very “robust” quorum sets—being able to tolerate up to $f = 2$ failures, which corresponds to a minimal blocking set of cardinality 3—the smallest blocking set of (\mathbf{V}, \mathbf{Q}) , $\{2\}$, actually has cardinality 1. Consider a failure of node 2. Node 0’s quorum set $(\mathbf{Q}(0))$ is not satisfiable anymore, so that 0 de-facto fails as well. With both 0 and 2 failed, node 1, being able to tolerate only $f = 1$ failures, becomes unsatisfiable as well. With three nodes having de-facto failed, none of the remaining nodes’ quorum sets can be satisfied anymore, so that (\mathbf{V}, \mathbf{Q}) loses quorum availability. Enabled through the “weak” quorum sets of nodes 0 and 1, the failure of 2 triggers what we would call a *cascading failure*. The liveness “buffer” of (\mathbf{V}, \mathbf{Q}) , as represented by its smallest blocking sets, is determined by the most easily dissatisfied nodes in its top tier.

We see a similar, although weaker effect with regards to minimal splitting sets. In the present example, there are fewer minimal splitting sets $\hat{S} \subset 2^{\mathbf{V}}$ than in an “ideal” FBAS of the same size (cf. *Ideal Open QSC* in Sec. 6.1) but all but one of them have the “ideal” cardinality 3 or a larger cardinality:

$$\hat{S} = \{\{1, 2\}, \{0, 1, 3\}, \{0, 1, 4\}, \{0, 2, 3\}, \{0, 2, 4\}, \{0, 3, 4\}, \{1, 3, 4, 5\}, \{2, 3, 4, 5\}\}$$

Note that unlike blocking sets that can compromise liveness for all nodes in an FBAS, splitting sets are usually more relevant to some nodes than they are to others. For example, the smallest splitting set of (\mathbf{V}, \mathbf{Q}) , $\{1, 2\}$, can potentially cause node 0 to diverge from the remainder of the network—this is likely a bigger problem for node 0 than for nodes $\{3, 4, 5\}$ which would remain “in sync”.

C Example analysis: Stellar network

As an example for the results obtainable using the proposed methodology and tooling, we will now present a short study into the Stellar FBAS [13]¹⁴. Our analysis methodology has furthermore been integrated into *Stellarbeat*¹⁵, a popular monitoring website for the Stellar network.

For the presented study, we obtain daily snapshots of the Stellar FBAS from Stellarbeat¹⁶, for the interval July 2019 – January 2022. From the same source, we also obtain data for allocating nodes, here individual network hosts running the

¹⁴ We maintain an interactive version of this study at: https://trudi.weizenbaum-institut.de/stellar_analysis/

¹⁵ <https://stellarbeat.io/>

¹⁶ Data from Stellarbeat was also used in previous academic studies such as [11].

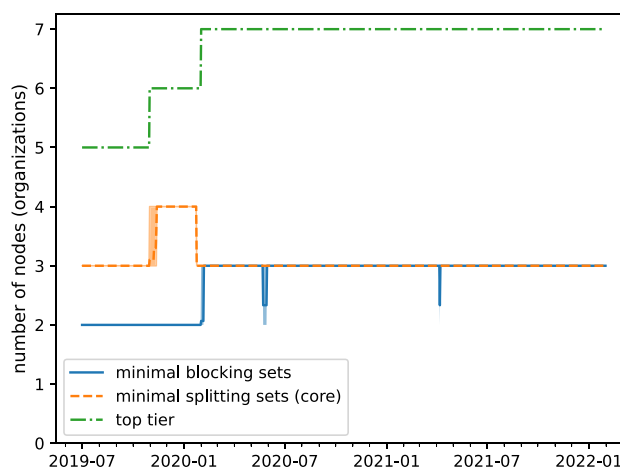


Fig. 5 Analysis results for daily snapshots of the Stellar network. For each presented FBAS snapshot, the plot charts the size of its top tier as well as the mean cardinalities of minimal blocking and minimal splitting sets, with area boundaries marking the cardinalities of the smallest and largest respective set

Stellar software, to the *organizations* they belong to. We use this data to merge nodes belonging to the same organization, so that nodes in the subsequent discussion represent distinct organizations as opposed to individual physical machines¹⁷. For maintaining the correctness of our results, we merge nodes in this way *after* completing the analyses. *Prior* to analysis, we filter out all nodes that are marked as inactive or induce one-node quorums (i.e., nodes v with a configuration such as $\mathbf{Q}(v) = \{v\}$; we assume that this represents an accidental misconfiguration). We furthermore restrict our minimal splitting sets analyses to a core subset of nodes for each FBAS snapshot, namely to the top tier and all nodes transitively referenced by top tier nodes’ quorum sets. Doing so gives us more informative aggregate results as forming a splitting set that affects only a few edge nodes is both significantly easier and less impactful than forming a splitting set that can cause top tier nodes to diverge. All analyses were performed using the algorithms and implementation introduced in Sec. 5. The results of our study are presented in Fig. 5.

The top tier of the Stellar network is growing monotonically through time in the studied interval, reaching 7 organizations in February 2020. The top tiers of most analyzed snapshots are symmetric and resemble (on the organizations level) a classical (non-nested) threshold-based quorum system. In Fig. 5, symmetric top tiers of such a type manifest themselves as data points in which the cardinalities

¹⁷ Nodes can also be merged based on other criteria, such as their country or ISP, revealing different threat scenarios. For example, for a Footnote 17 continued snapshot of the Stellar FBAS from November 2020, we determine that a certain large cloud hosting provider forms a blocking set—i.e., has the power to unilaterally compromise liveness.

of all minimal blocking sets are identical, as are the cardinalities of all minimal splitting sets. During February 2020, the top tier grew by one organization, disturbing the symmetry for a few days. However, eventually all top tier nodes included the new organization into their quorum sets. This adaptation suggests that top tier nodes might be reacting to each others' decisions and actively strive towards a symmetric configuration, as proposed in Sec. 6.4. Furthermore, the thresholds of top tier quorum sets appear to be chosen based on a 67% logic (balancing liveness and safety risks), as do most example policies we discuss in Sec. 6.

References

- Bracciali, A., Grossi, D., de Haan, R.: Decentralization in open quorum systems: Limitative results for Ripple and Stellar. In: 2nd International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2020), pp. 5:1–5:20. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021)
- Cachin, C., Tackmann, B.: Asymmetric distributed trust. In: 23rd International Conference on Principles of Distributed Systems (OPODIS 2019), pp. 7:1–7:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2020)
- Cachin, C., Zanolini, L.: Asymmetric asynchronous byzantine consensus. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology, pp. 192–207. Springer (2021)
- Castro, M., Liskov, B., et al.: Practical Byzantine fault tolerance. In: Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI), pp. 173–186. USENIX, New Orleans, Louisiana, USA (1999)
- Douceur, J.R.: The Sybil attack. In: Peer-to-peer Systems, pp. 251–260. Springer, Berlin, Heidelberg (2002)
- Ford, B., Böhme, R.: Rationality is self-defeating in permissionless systems (2019). [arxiv:1910.08820](https://arxiv.org/abs/1910.08820)
- Gallo, G., Longo, G., Pallottino, S., Nguyen, S.: Directed hypergraphs and applications. *Discrete appl. math.* **42**(2–3), 177–201 (1993)
- García-Pérez, Á., Gotsman, A.: Federated Byzantine quorum systems. In: 22nd International Conference on Principles of Distributed Systems (OPODIS 2018), pp. 17:1–17:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018)
- García-Pérez, Á., Schett, M.A.: Deconstructing Stellar consensus. In: 23rd International Conference on Principles of Distributed Systems (OPODIS 2019), pp. 5:1–5:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2020)
- Gaul, A., Khoffi, I., Liesen, J., Stüber, T.: Mathematical analysis and algorithms for federated Byzantine agreement systems (2019). [arxiv:1912.01365](https://arxiv.org/abs/1912.01365)
- Kim, M., Kwon, Y., Kim, Y.: Is Stellar as secure as you think? In: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 377–385. IEEE, Stockholm, Sweden (2019)
- Lachowski, Ł.: Complexity of the quorum intersection property of the federated Byzantine agreement system (2019). [arxiv:1902.06493](https://arxiv.org/abs/1902.06493)
- Lokhava, M., Losa, G., Mazières, D., Hoare, G., Barry, N., Gafni, E., Jove, J., Malinowsky, R., McCaleb, J.: Fast and secure global payments with Stellar. In: Proceedings of the 27th ACM Symposium on Operating Systems Principles (SOSP '19), pp. 80–96. ACM, New York, NY, USA (2019)
- Losa, G., Gafni, E., Mazières, D.: Stellar consensus by instantiation. In: 33rd International Symposium on Distributed Computing (DISC 2019), pp. 27:1–27:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2019)
- Malkhi, D., Reiter, M.: Byzantine quorum systems. *Distributed comput.* **11**(4), 203–213 (1998)
- Mazières, D.: The Stellar consensus protocol: A federated model for internet-level consensus (2015). <https://stellar.org/papers/stellar-consensus-protocol.pdf>
- Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008). <http://nakamotoinstitute.org/bitcoin/>
- Ndolo, C., Henningsen, S., Florian, M.: Crawling the MobileCoin quorum system (2021). [arxiv:2111.12364](https://arxiv.org/abs/2111.12364)
- Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. Tech. rep, Stanford InfoLab (1999)
- Stathakopoulou, C., David, T., Vukolić, M.: Mir-BFT: High-throughput BFT for blockchains (2019). [arxiv:1906.05552](https://arxiv.org/abs/1906.05552)
- Tarjan, R.: Depth-first search and linear graph algorithms. *SIAM j. on comput.* **1**(2), 146–160 (1972)
- Yin, M., Malkhi, D., Reiter, M.K., Gueta, G.G., Abraham, I.: HotStuff: BFT consensus with linearity and responsiveness. In: Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC '19), pp. 347–356. ACM, New York, NY, USA (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.