Constellation: Peer-to-Peer Overlays for Federated Byzantine Agreement Systems

Giuliano Losa[1], Yifan Mao[2], Shaileshh Bojja Venkatakrishnan[2], and Yunqi Zhang[2]

[1] Stellar Development Foundation, USA
[2] The Ohio State University, USA

Abstract. A federated Byzantine agreement (FBA) system is a permissionless system in which each participant declares unilateral agreement requirements that collectively determine a set of quorums. The resulting quorums can be used in a consensus algorithm (such as the Stellar Consensus Protocol) to obtain a permissionless blockchain system without resorting to proof-of-work or proof-of-stake. Like most permissionless systems, FBA systems must rely on a secure peer-to-peer overlay network for communication between network nodes, yet this topic has received little attention in the FBA setting.

In this paper, we focus on the problem of connecting the nodes of an FBA system to each other in order to obtain an overlay topology that securely and efficiently supports gossip protocols, a popular class of protocols for disseminating data over blockchain overlays. We present Constellation, an algorithm that computes an overlay topology whose resilience matches the resilience of the FBA quorum system. Constellation minimizes node degree, which reduces redundant traffic in gossip protocols, and it ensures a maximal diameter of 2, which is crucial for low-latency consensus.

1 Introduction

A federated Byzantine agreement system (FBA system, or FBAS) [33,28] is an open, permissionless distributed system in which nodes intend to reach agreement, e.g. on the next block of a blockchain, subject to *agreement requirements* that each node is free to choose for itself. Roughly speaking, each node declares its agreement requirements in the form of a set of so-called *quorum slices*, each of which is a set of other nodes, with the intent to never agree to anything unless at least one of its quorum slices unanimously agrees to it. Under such constraints, a set of nodes Q can reach agreement only if it Q satisfies the agreement requirements of all its members (e.g. if Bob requires agreement with Alice and Alice requires agreement with Bob, then they need each other to agree to anything). Such a set Q forms what we call a *quorum*. Thus an FBAS gives rise to a set of quorums, which can be used in a quorum-based Byzantine Fault-Tolerant (BFT) consensus algorithm, in a way that is decentralized and resilient to Sybil attacks [15] without resorting to proof-of-stake or proof-of-work.

A fundamental challenge in an FBA system, and most blockchain systems, is the design of an efficient and fault-tolerant peer-to-peer (p2p) overlay network. In a p2p overlay, nodes establish connections between each other through the Internet, and those connections are then used by a protocol—often a gossip protocol [14,22]—to disseminate information throughout the system. The characteristics of the p2p overlay, e.g. its diameter and its degree distribution, heavily impact overall system performance; moreover, failure to keep the overlay connected may lead to a loss of liveness for parts of or for the whole system. While several works study consensus or reliable broadcast protocols for FBA systems [33,28,29,18,17,9] or related models [10,24,11], the problem of building an efficient p2p topology for the FBA model has received little attention.

In practice, a popular approach to constructing p2p overlays is for each node to connect to a random sample of the peers it knows about. The Stellar network [3]—currently the largest deployed FBA system—uses such a random approach. While this is simple to implement and robust in benign scenarios, it is hard to give any guarantees when under Byzantine or Sybil attacks.

In an FBA system, we would like the overlay to remain functional despite failures as long as the failures are not severe enough to preclude meeting the agreement requirements of the nodes that remain well-behaved. Otherwise, the p2p overlay would cripple the liveness guarantees of the BFT consensus algorithm running on top. A random overlay cannot guarantee this property unless nodes connect to an excessive number of peers, which could severely impact system performance. For example, as of January 3rd, 2025, the subgraph formed by the top-tier nodes[3] of the Stellar network has an average degree of 4.3; unfortunately, this allows failure scenarios in which the well-behaved nodes form a quorum, and thus they should be able to make progress, but progress is impossible because the overlay becomes disconnected.

In this work, we investigate how to build a efficient p2p overlay networks for FBA systems. We consider two key requirements. First, we require the well-behaved nodes to remain connected to each other as long as they form a quorum. In this case, we say that the overlay is *FBA-resilient*.

A naïve approach is for a node to connect with all of the nodes it requires agreement from[4], but this creates too many connections. A better approach might be for a node to choose neighbors such that there is a connection to at least one node from each of its quorum slices; this ensures at least one available, well-behaved neighbor. However, we aim for the stronger requirement of having a path (of well-behaved nodes) between any two well-behaved nodes.

Second, we require the overlay to have as low a diameter and node degrees as possible. A low diameter ensures messages published in the network are received by the nodes quickly, which is crucial for latency-sensitive messages like votes in a consensus algorithm. Similarly, a low node degree reduces bandwidth over-utilization, which in turn minimizes congestion and its associated delays. While the choice of gossip protocols (e.g., pull-based vs. eager push §2) used

[3] See Section 2.2.

[4] Clearly some of the connection requests may fail, if the requested nodes are offline.

for messages dissemination does impact the efficiency of message dissemination in the network, the p2p topology is a more fundamental parameter on which the gossip protocols' performance depends. In this work, we consider only the problem of topology optimization and not gossip-protocol design.

To build overlays satisfying the two requirements above, we present Constellation, a decentralized blockchain-assisted topology-construction algorithm for computing an overlay with a low diameter and low average node degree. Importantly, Constellation exploits the agreement requirements of nodes for neighbor selection, resulting in an overlay that is resilient to failures and malicious attacks: the overlay remains connected provided well-behaved nodes form a quorum.

Similarly to how proof-of-stake systems track stakers and their stake on chain, Constellation tracks nodes and their agreement requirements on chain. Given global agreement on the node's agreement requirements, as provided by the blockchain, each node locally applies the Constellation overlay-building algorithm, which is deterministic, and obtains a graph that determines which nodes it should connect to or accept connections from. However, even with global knowledge of the agreement requirements of each node, finding an optimal overlay topology that satisfies our requirements is a hard combinatorial problem.

To tractably compute an overlay, Constellation first conservatively approximates each node's quorum slices using a family of sets that can be described succinctly using simple thresholds (in practice, we expect quorum slices to be specified in this form already, as is the case in the Stellar network). Then, to avoid combinatorial explosion, Constellation searches for an optimal overlay that follows a pre-defined template. This template consists of assigning nodes to clusters whose membership depends on the threshold parameters of their members, and finally creating inter-node edges that result in high interconnection within clusters and more sparse connections across clusters. If all nodes have the same threshold, the connectivity between any two clusters forms a bipartite matching, with each cluster being a partite.

The graphs obtained by Constellation bear resemblance to the Cartesian product of complete graphs, whose strong connectivity properties have been extensively studied in the literature [20,34,30,25,40]. In fact, in the main technical result of the paper, we leverage those existing results on the connectivity of Cartesian products of graphs to prove that, when all nodes have the same threshold, Constellation produces an FBA-resilient overlay. Moreover, the diameter of a Constellation overlay is 2 by construction, and we show that it remains below 3 even under a significant fraction of failures. Finallly, We also show that the average node degree in Constellation is near-optimal.

We have evaluated Constellation using simulations and on a real-world testbed running stellar-core [2], a production implementation of a blockchain over an FBA system. Our results show that overlays generated by Constellation achieve higher transaction throughput compared to random overlays while also guaranteeing connectivity as long as the FBA agreement requirements can be met.

Our work is the first to propose a structured p2p overlay for blockchains that is secure under the same failure assumptions as the blockchain's consen-

sus protocol. In a proof-of-stake setting, Coretti et al. [13] as well as Liu-Zhang et al. [27,26] propose to construct random graph overlays in which each node's degree is proportional to the amount of stake it has. It is unclear how such a construction can be adapted to an FBA system with subjective agreement requirements. Overlays using a distributed hash table (DHT; e.g., KadCast [38] that uses the Kademlia DHT [32]) are common, but they ignore the trust preferences of nodes.

Software and instructions to reproduce most of our evaluation results (both simulations and testbed experiments) is available online [4].

2 Background: Federated Byzantine Agreement Systems

In this paper we are interested in overlays for federated Byzantine agreement systems (FBAS) [33,28], as most prominently used in the Stellar network [3]. In this section, we briefly cover both theoretical and practical aspects of FBA systems that are necessary to understand the rest of the paper.

2.1 Federated Byzantine Agreement in Theory

An FBA system consists of a set N of nodes (which, in a permissionless deployment, is unknown to the nodes) in a network and which may be either well-behaved or Byzantine, and where nodes are trying to reach an agreement on some value (typically, the next block in a blockchain). Well-behaved nodes follow their assigned algorithm and are available (i.e. responsive), while Byzantine nodes may behave arbitrarily, which models nodes controlled by an attacker.

Each node declares agreement requirements by specifying a set of sets of nodes called *quorum slices*, with the intent not to agree to anything unless at least one of its quorum slices unanimously agrees to it. This means that, for a set of nodes Q to be able to reach agreement even if the rest of the system is Byzantine (e.g. if $N \setminus Q$ is not available), Q must be such that, for every node n in Q, n has a quorum slice that is a subset of Q. In other words, Q can satisfy the agreement requirements of all its members. We call such sets quorums:

Definition 1 (Quorum). *A set Q is a quorum when every node in Q has a quorum slice that is a subset of Q.*

Under the assumption that all quorums suitably intersect (see [33,28,29] for precise definitions) and that the set of well-behaved nodes forms a quorum, consensus algorithms like the Stellar Consensus Protocol (SCP) [33,28] or the algorithm of [29] solve consensus under eventual synchrony[5].

In this work, given a FBA system, our goal is to create an overlay that guarantees connectivity and diameter 2 in all situations under which a consensus algorithm can be expected to make progress, i.e. under the assumption that the set of well-behaved nodes forms a quorum.

[5] Technically, to ensure termination, SCP additionally assumes that, eventually, Byzantine nodes stop interfering

2.2 Federated Byzantine Agreement in Practice

We expect nodes in a real FBAS to belong to known, real-world organizations that run them, with a single operator per organization, and for most node operators to configure their node's quorum slices using a simple threshold of organizations. We formalize this assumption in Definition 2 with the notion of a regular FBAS, which we illustrate in Example 1.

Definition 2 (Regular FBAS). *A regular FBAS consists of a set of organizations $\mathcal{O}$ each running disjoint sets of nodes. For each organization $O \in \mathcal{O}$, let $n(O)$ be the set of nodes run by organization O, and let $N = \cup_{O \in \mathcal{O}} n(O)$ be the set of all nodes. We assume that the operator of the nodes belonging to an organization O chooses a set of organizations $\mathcal{T}_O \subseteq \mathcal{O}$ to trust, called O's universe, and an integer threshold $0 < t_O \leq |\mathcal{T}_O|$. This determines the quorum slices of every node $n \in n(O)$ as follows: n's quorum slices are the sets obtained by (a) picking a number t_O of organizations among $\mathcal{T}_O$ and (b), for each picked organization O, by picking a strict majority of O's nodes.*

Example 1. Suppose that there are 4 organizations O_i, for i from 1 to 4, each running 3 nodes $n(O_i) = \{n_i^a, n_i^b, n_i^c\}$, and suppose that, for all i, $T_{O_i} = \{O_1, O_2, O_3, O_4\}$ and $t_{O_i} = 3$. Then, each node has a set of quorum slices obtained by picking 3 organizations out of the four and then picking 2 nodes out of 3 (i.e. a strict majority) of each picked organization. For instance, $\{n_1^a, n_1^b, n_2^b, n_2^c, n_4^a, n_4^c\}$ is a quorum slice, but $\{n_1^a, n_2^b, n_2^c, n_3^a, n_3^b, n_3^c\}$ is not. Finally, in this example, quorum slices and minimal quorums coincide.

In practice, we observe that most nodes in the Stellar network follow the structure above. The universe $\mathcal{T}_O$ and threshold t_O associated with an organization O correspond to what is called a *quorum set*[6] in Stellar's terminology[7].

Finally, when all organizations in a regular FBAS have the same universe, we say that the FBAS is a *single-universe* regular FBAS, and if all organizations in a symmetric FBAS have the same threshold, we say that the FBAS is *symmetric*.

Top-tier and second-tier nodes. In practice in the Stellar network, we observe that a small subset of the nodes can be classified as top-tier nodes, while we call other nodes second-tier nodes. Top-tier nodes only require agreement from other top-tier nodes while second-tier nodes require agreement from the top-tier nodes and, sometimes, other second-tier nodes.

The top-tier nodes in the Stellar network are nodes run by organizations, such as the Stellar Development Foundation, the infrastructure-provider Blockdaemon, or the investment firm Franklin Templeton, which are well-known in

[6] Technically, Stellar's quorum sets can have a more refined structure, but it is rarely used in practice.

[7] The name "quorum set" used by Stellar is unfortunate, as a "quorum set" is not a set of quorums but instead specifies a set of quorum slices.

the Stellar community. As of January 3rd, 2025, there were 23 top-tier nodes belonging to 7 organizations and 458 second-tier nodes[8]. Due to how communities in the real-world often coalesce around a few key players, we expect that most FBA systems in practice will have a similar structure.

The fact that the large number of second-tier nodes require agreement from the top-tier nodes but not vice-versa poses a problem from the point of view of building an overlay that guarantees node can communicate with who they require agreement from. That is because the small top-tier would need to be densely connected to the much larger rest of the network, meaning that top-tier nodes would have a very large degree. Unfortunately, this is intrinsic to our requirements and there is no way around this problem. Worse, in an open system like the Stellar network, nothing can prevent a Sybil adversary from creating arbitrarily many second-tier nodes to further overload the top-tier nodes.

Faced with this conundrum, we suggest using a best-effort random-overlay approach to connect second-tier nodes to each other and to top-tier nodes, and to give resilience guarantees only to top-tier nodes. In the rest of the paper we follow this approach, and the novel aspects of our work concern building overlays connecting the top-tier nodes together. We note that Liu-Zhang et al. face a similar issue in the context of proof-of-stake [26, Section 5]. They observe that nodes may exist in the system that have no stake (in some sense, those cannot be trusted because they have nothing at stake) but that nevertheless need to be connected to the overlay because legitimate users depend on them. They conclude that no guarantees can be given to those nodes without additional assumptions limiting Sybil attacks.

3 Model and Problem Formulation

We consider a regular FBAS $\mathcal{F}$ (Definition 2) consisting of a set of nodes N, also called validators, and we define the set of top-tier nodes TT as the union of all minimal quorums of the FBAS. The set of top-tier nodes also determines the set of top-tier organizations TO, where an organization is top tier when all its nodes are top tier nodes. When an organization or node is not top tier, we say it is second tier (moreover, note that, in a regular FBAS, either all nodes of an organization are top tier or none are). We further assume that no second-tier organization appears in the universe of any top-tier organization. This means that the top-tier TT forms a regular FBAS on its own.

We say that the top-tier TT tolerates the failure of a set of nodes B when $TT \setminus B$ is a quorum (equivalently, every top-tier node not in B has a quorum slice disjoint from B). We are concerned with computing an *FBA-resilient overlay*

[8] This data was obtained using surveying functionality built into stellar-core [2] (Stellar's blockchain implementation), and communicated to the authors privately by the Stellar Development Foundation. Note that `https://stellarbeat.io` reports less than 200 nodes, as of January 2024, and this is because many nodes do not accept inbound overlay connections from its crawler.

over TT, which is a graph whose nodes are the members of TT and that satisfies the following definition.

Definition 3 (FBA-Resilient Overlay). *We say that an overlay over a set of nodes is* FBA-resilient *when, if we remove from the graph a set of nodes B such that the FBAS tolerates the failure of B, the graph remains connected.*

The problem that we address with Constellation is to connect the top-tier nodes to form an overlay that (a) is FBA-resilient, (b) minimizes the average and maximum degree of the nodes, and (c) has diameter of at most 2. Moreover, when taking the union with a random overlay connecting the second-tier nodes (e.g. an Erdős–Rényi graph spanning all the nodes), the combined overlay must allow a consensus algorithm to achieve high performance in practice.

In the next section (Section 4), we present the Constellation algorithm, which, given a regular FBAS, computes an efficient and FBA-resilient overlay. Then, in Section 5, we discuss how to deploy this algorithm to obtain a practical solution to create and maintain an overlay connecting the nodes of a FBA system implementing a blockchain, like the Stellar network.

4 The Constellation Algorithm

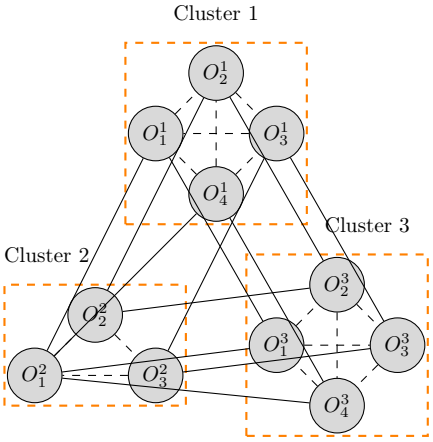
The Constellation algorithm takes as input a regular FBAS (Definition 2) and computes an overlay, which is a graph over the nodes of the FBAS. The first step in the Constellation algorithm is to conservatively simplify the FBAS by setting $T_O = \mathcal{O}$ for each organization O, without modifying t_O, thereby obtaining a single-universe, regular FBAS. Note that this transformation has the property that, if the well-behaved nodes form a quorum in the original FBAS $\mathcal{F}_1$, they also form a quorum in the new FBAS $\mathcal{F}_2$, and so if an overlay is FBA-resilient in $\mathcal{F}_2$ then it is also FBA-resilient in $\mathcal{F}_1$.

Next, the Constellation algorithm determines a graph over the nodes of the (now single-universe) FBAS by instantiating the *Constellation Topology Template*. To instantiate the template, the Constellation algorithm performs a brute-force search over the parameters of the template in order to find an instantiation that minimizes the average degree of the resulting overlay.

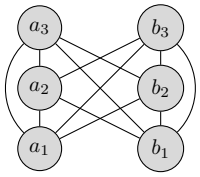
In the rest of this section, we present the Constellation Topology Template (Section 4.1) and we prove that the construction results in a diameter of 2. Then, in Section 4.2, we show how to instantiate the template when the FBAS is symmetric and we prove that the resulting overlay has near-optimal average degree, is FBA-resilient, and remains of diameter 3 even under a large number of failures. We discuss how we search for optimal template parameters when the FBAS is not symmetric in Appendix F.

4.1 The Constellation Topology Template

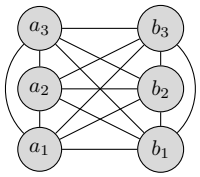
Let $G(N, E)$ denote the undirected graph of the overlay topology generated by Constellation. The high-level structure of G can be understood as a hyper-



(a) Overlay construction for a symmetric system of 11 organizations with a threshold of $\lceil 2 \cdot 11/3 \rceil = 8$. Dotted and solid edges correspond to the two inter-organization connection patterns depicted in Figures 1b and 1c, respectively.



(b) Node-level intra-cluster connections (dotted edges) between two organizations A and B with three nodes each (a_1, a_2, and a_3 and b_1, b_2, and b_3, respectively).



(c) Node-level inter-cluster connections (solid edges) between two organizations A and B with three nodes each (a_1, a_2, and a_3 and b_1, b_2, and b_3, respectively)

Fig. 1: Example of the cluster-based network structure produced by Constellation. The two types of intra-organization edges (dotted and solid) appearing on Figure 1a are realized at the node level as shown on Figures 1b and 1c.

graph[9] H defined on $\mathcal{O}$. For every two organizations O_1, O_2, we say there is a hyperedge (O_1, O_2) between O_1 and O_2 if there are edges between the nodes in O_1 and O_2 in G. Conversely, there is no hyperedge between O_1 and O_2 if none of the nodes in O_1 are connected to any of the nodes in O_2 in G.

Constellation partitions the set of organizations $\mathcal{O}$ into a set of clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$ where $k > 0$ is a parameter. Within each cluster $\mathcal{C}_i$, the organizations form a complete hypergraph, i.e., $(O_1, O_2) \in H$ for all $O_1, O_2 \in \mathcal{C}_i$. Additionally, an organization $O \in \mathcal{C}_i$ also has a hyperedge to at least one organization in each of the other clusters $\mathcal{C}_j, j \neq i, j \in \{1, \ldots, k\}$. This last requirement can be satisfied in many ways. However, to avoid excessive hyperedges on a single organization, hyperedges between two clusters $\mathcal{C}_i, \mathcal{C}_j$ are distributed evenly (as much as possible) among the constituent organizations: If $\{O_1, \ldots, O_{|\mathcal{C}_i|}\}$ and $\{O'_1, \ldots, O'_{|\mathcal{C}_j|}\}$ are the organizations in $\mathcal{C}_i$ and $\mathcal{C}_j$ respectively, we have $(O_l, O'_{l \mod |\mathcal{C}_j|}) \in H$ for all $l \in \{1, \ldots, |\mathcal{C}_i|\}$ and $(O_{l \mod |\mathcal{C}_i|}, O'_l) \in H$ for all $l \in \{1, \ldots, |\mathcal{C}_j|\}$.

Note that the hypergraph is parameterized by the number of clusters k and the assignment of organizations to clusters. These are the parameters we will

[9] By hypergraph we simply mean a graph defined on organizations and not nodes.

later try to optimize. Figure 1a shows an example of Constellation hypergraph when there are 11 organizations and 3 clusters.

From Hyperedges to Node-Level Edges Consider any two organizations O_1, O_2 that belong to the same cluster. To realize the hyperedge connecting them, we construct a bipartite graph between O_1 and O_2 such that (1) each validator $v \in O_i$ is connected to at least half of the validators in $O_j, j \neq i, j \in \{1, 2\}$, and (2) the degrees of the nodes in each partite are evenly distributed (as much as possible). Letting $\{v_1, \ldots, v_{|O_1|}\}$ and $\{v'_1, \ldots, v'_{|O_2|}\}$ be the validators in O_1 and O_2, respectively, we include edges $(v_i, v'_{i \mod |O_2|}), (v_i, v'_{i+1 \mod |O_2|}), \ldots, (v_i, v'_{i+\lfloor |O_2|/2 \rfloor \mod |O_2|})$ in the bipartite graph for all $i \in \{1, \ldots, |O_1|\}$. Similarly, we include edges $(v_{i \mod |O_1|}, v'_i), (v_{i-1 \mod |O_1|}, v'_i), \ldots, (v_{i-\lfloor |O_1|/2 \rfloor \mod |O_1|}, v'_i)$ in the bipartite graph. For organizations O_1, O_2 that belong to different clusters but have an hyperedge between them, we let each validator $v \in O_i$ have an edge to all the validators in $O_j, j \neq i, j \in \{1, 2\}$. Each validator in an organization has an edge to all the other validators in the organization. Figures 1b and 1c illustrate the connections between two organizations when they belong to the same cluster and different clusters, respectively. Our first result is that, regardless of how the topology parameters are selected, the graph we obtain has a diameter of 2:

Proposition 1. *Regardless of how organizations are partitioned into clusters, the graph $G(N, E)$ constructed by Constellation has a diameter of 2.*
(Proof in Appendix B)

4.2 Ensuring Connectivity and Low Diameter under Failures

In Section 4.1 above we have presented the general Constellation Topology Template. However, we did not specify how to determine how many clusters to create and how to assign organizations to clusters. To ensure the Constellation graph G is FBA-resilient (Definition 3), i.e. that it remains connected and of low diameter even under any node failures tolerated by the FBAS, it is important for organizations to have hyperedges to as many other organizations as possible[10]. However, it is also important to minimize node degree.

We propose to partition the organizations so as to ensure that, in the resulting overlay, each node has at least one connection to each of its quorum slices. Despite this seemingly weak (local) requirement, we show in the rest of this section that, in the case of a symmetric FBAS (i.e. when all organizations have the same threshold), the obtained overlay is FBA-resilient (i.e. global connectivity is guaranteed under maximal failures tolerated by the FBAS). Moreover, we prove that (again, for a symmetric FBAS) the resulting overlay has near-optimal average degree and remains of diameter 3 even under a large number of failures.

Recall that Constellation first conservatively approximates the regular FBAS formed by the top-tier nodes to a single-universe, regular FBAS, where each

[10] This would be achieved by using singleton clusters only, which would result in a complete hypergraph

validator $v \in N$ has the universe $\mathcal{T}_v = \mathcal{O}$ (the set of all top-tier organizations) and a threshold $t_v = t_O$, where O is v's organization, with $0 < t_O \leq |\mathcal{O}|$.

For an organization $O \in \mathcal{O}$ with a threshold of t_O, we require O to be part of at least $|\mathcal{O}| - t_O + 1$ hyperedges in H. This ensures O has a hyperedge to at least one available organization, even when $|\mathcal{O}| - t_O$ organizations are unavailable. Similarly, whenever $(O, O') \in H$ for any organization $O' \in \mathcal{O}$, we require validators $v \in O$ to form edges with at least half of the validators in O'. This ensures v has an edge to at least one available validator within each available organization.

A key challenge now is computing the total number of clusters and assignments of organizations to those clusters such that the hypergraph degree requirements described above are satisfied, but without overly increasing the degree. Assuming that all validators use the same threshold t with $\frac{2}{3}|\mathcal{O}| < t \leq |\mathcal{O}| + 2 - 2\sqrt{|\mathcal{O}|}$ and $\mathcal{T}_v = \mathcal{O}\ \forall v \in N$, computing the cluster assignments is straightforward[11]: we compute the number of clusters k from the inequality[12]

$$k + \frac{|\mathcal{O}|}{k} - 1 \geq |\mathcal{O}| - t + 1, \text{ leading to} \tag{1}$$

$$k = \left\lfloor \frac{2 + |\mathcal{O}| - t}{2} - \frac{\sqrt{4 + |\mathcal{O}|^2 - 4t - 2|\mathcal{O}|t + t^2}}{2} \right\rfloor. \tag{2}$$

Assuming all clusters have the same number of organizations, the left-hand side of inequality (1) above counts the number of hyperedges an organization has outside its cluster (k-1) and within its cluster ($|\mathcal{O}|/k - 1$), plus one implicit self hyperedge, while the right-hand side is the desired degree requirement. In practice, if the number of organizations is not exactly divisible by k, we let each cluster $\mathcal{C}_i$ have $\lfloor |\mathcal{O}|/k \rfloor$ organizations for $i = 1, \ldots, k-1$ and the k-th cluster $\mathcal{C}_k$ have $|\mathcal{O}| - (k-1) * \lfloor |\mathcal{O}|/k \rfloor$ organizations. The resulting topology not only has a diameter of 2 (Proposition 1) but also has near-optimal degree:

Theorem 1. *In the symmetric case with $t_v = t, \mathcal{T}_v = \mathcal{O}$ for all $v \in N$, each cluster having the same number of organizations, and each organization having the same number of validators, the validator degree achieved by Constellation exceeds the optimal degree by a factor at most $2 + \frac{2}{k^2 f} + \frac{1}{\sqrt{|\mathcal{O}|} - 1/2}$ for $\frac{2|\mathcal{O}|}{3} \leq t < |\mathcal{O}| + 2 - 2\sqrt{|\mathcal{O}|}$ and by a factor at most $\frac{6}{\sqrt{|\mathcal{O}|} f}$ for $|\mathcal{O}| + 2 - 2\sqrt{|\mathcal{O}|} \leq t \leq |\mathcal{O}|$, where $f = \frac{|\mathcal{O}| - t}{|\mathcal{O}|}$ denotes the maximum fraction of organizations that can fail. (Proof in Appendix C)*

Figure 7 (Appendix C) suggests that, for t close to $2|O|/3$, the optimality factor of Theorem 1 quickly stabilizes under 3 as $|O|$ increases.

[11] $t < \frac{2}{3}|\mathcal{O}|$ is out of scope since it produces disjoint quorums.

[12] For $|\mathcal{O}| + 2 - 2\sqrt{|\mathcal{O}|} < t < |\mathcal{O}|$, any $k > 0$ satisfies inequality (1). To minimize the degree, we have $k = \lfloor \sqrt{|\mathcal{O}|} \rfloor$ for this case.

When validators have different thresholds, computing the cluster assignments to minimize degree is not straightforward but can be done algorithmically. We discuss this in Appendix F.

Next, for the symmetric case (with identical thresholds), Theorem 2 shows that the Constellation topology remains connected under the maximal failures of $|\mathcal{O}| - t$ organizations allowed under the FBA model.

Theorem 2. *For symmetric thresholds $t_v = t, \mathcal{T}_v = \mathcal{O}$ for all $v \in N$, $t \geq 2|\mathcal{O}|/3$, each cluster having the same number of organizations, and each organization having the same number of validators, the Consthe number of organizations that must be removed to disconnect the Constellation network is at least $|\mathcal{O}| - t + 1$. (Proof in Appendix D)*

Finally, even in non-symmetri cases, Constellation also ensures a diameter of 3 even under a significant number of failures.

Proposition 2. *For k clusters, with each cluster having the same number of organizations, the diameter of the Constellation network is at most 3 even under failure of $|\mathcal{O}|/k + k - 2$ organizations.*
(Proof in Appendix E)

Combined with the result of Theorem 2, Proposition 2 shows that the diameter of Constellation is near optimal even when the number of failed nodes is close to that needed for disconnecting the network.

5 Deployment of Constellation in an FBA Blockchain

As we have seen, the Constellation algorithm takes as input a regular FBAS, determines its top-tier (i.e. the union of all minimal quorums), and then computes an overlay connecting the top-tier nodes.

To deploy Constellation in an FBA blockchain network, we use the blockchain to store the configuration of each node (in practice, nodes would submit special configuration-registration transactions to the blockchain which, when executed, update a shared data-structure indicating configuration of each registered node). As long as a superset of the top-tier registers their configuration, this guarantees that all nodes can compute what the top-tier is. Moreover, since the Constellation algorithm is deterministic, every node can then compute the Constellation overlay locally and arrive at the same result. Finally, nodes establish and accept connections according to the output of the Constellation algorithm. If a node receives an unexpected connection request, not prescribed by the Constellation algorithm, it can simply reject it.

In the Stellar network, nodes typically execute a new transaction set (i.e. a block) every 5 seconds, and changes to the configurations of top-tier node are comparatively infrequent (having changed only a handful of times in the last 3 years). To strike a balance between adaptivity and load on the network, we propose to recompute the overlay topology daily. Thanks to Constellation's resilience guarantees, the system will remain operational as long as, within a

day, the set of nodes S that fail or leave the system is a set that the FBAS tolerates the failure of, according to their configuration at the beginning of the day. Note that this is similar to how proof-of-stake blockchains manage the list of stakers and their stake on chain. Moreover, recent work on overlay networks for proof-of-stake systems [13,27,26] also relies on the list of stakers and their stake, which is maintained on the blockchain, to derive overlay connections.

Since this scheme relies on the blockchain to set up the overlay, and the blockchain relies on the overlay to run, we need to bootstrap the system in some way. Moreover, we need to connect second-tier nodes, which the Constellation algorithm does not handle. An easy way to do this is to bootstrap the system with the same random-connections scheme used today in the Stellar network. Concretely, we can instruct nodes to accept some fixed maximal number of connection requests on top of the connections prescribed by Constellation, and we can prescribe newly joining nodes to establish a fixed number of random connections to bootstrap their inclusion in the overlay (after they have obtained a list of node by contacting at least one existing node provided by their node operator). On top of allowing to bootstrap the system, this also allows nodes to join in the middle of the daily reconfiguration period. However, this random scheme is best effort and gives no guarantees.

Appendix G discusses several attack vectors, along with mitigations, that arise in deployments of Constellation following the scheme described above.

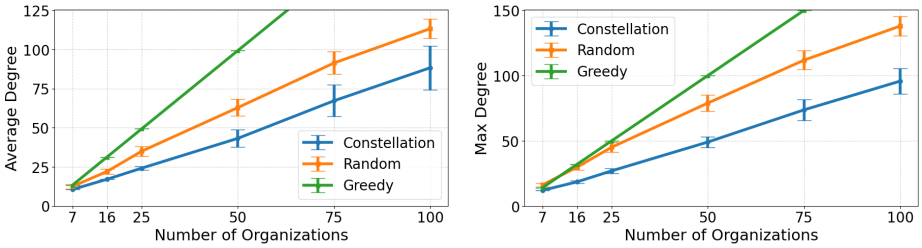
6 Evaluation

In this section we evaluate Constellation empirically against random and greedy overlay-building strategies, both in simulation and on a real-world testbed on Amazon EC2. We use random and greedy overlay-building strategies as our baselines because existing blockchain p2p overlay configurations (e.g., in Ethereum [5] or Bitcoin [21]) do not try to match the fault-tolerance of the consensus protocol running on top and, therefore, are hardly comparable to our approach.

6.1 Simulations on Synthetic Networks

Degree and Connectivity Analysis. In this first series of experiments, we consider a number of organizations ranging from 7 to 100, each consisting of 3 nodes, and forming a single-universe, regular FBAS. For each number of organizations, we conduct experiments in which we assign organizations random FBA thresholds between $|O|/2$ and $5|O|/6$, and we compare the maximum and average degree achieved with 3 different algorithms: a random algorithm, a greedy algorithm, and Constellation.

In the random algorithm, we let each node pick k neighbors uniformly at random and we repeat the experiment, each time incrementing k, until we obtain an overlay that has diameter 2 and that is resilient to the failures that must be tolerated according to the node's FBA thresholds.



(a) Average degree (b) Maximum degree

Fig. 2: Overlay degrees achieved with the Constellation algorithm, the greedy algorithm, and the random algorithm. Errors bars show standard deviation.

In the greedy algorithm, we first build a graph over organizations , starting with an empty graph, by repeatedly sorting the organizations in a list O_1, O_2, ..., $O_{|\mathcal{O}|}$ in descending order number of connections to other organizations still needed (recall that the minimum number of connections required by a quorum set $(\mathcal{O}, t_O)$ is $c_O = |\mathcal{O}| - t_O + 1$) and then, letting c_1 be the number of additional connections needed by O_1, we connect O_1 to the first c_1 organizations in the list that O_1 is not yet connected to. We break the loop when all organizations have at least their required number of connections. Then, we make node-to-node connections: for each edge (O_1, O_2) in the hypergraph, we use the pattern of Figure 1c, and we connect the 3 nodes of each organization using 2 edges. Finally, if the diameter of the overlay is greater than 2, we repeatedly connect the most distant pairs of nodes until the diameter becomes 2.

The results appear in Figure 2. We can see that Constellation consistently achieves better average and maximum degrees than the random and greedy strategy, and that the gap increases as the number of organizations increases. The greedy strategy is clearly not competitive and, at 100 organizations, Constellation achieves an average degree roughly 22% lower than the random algorithm and a max degree roughly 30% better.

Robustness Analysis To evaluate the robustness of networks using Constellation, we generate random FBA systems and simulate node failures by randomly picking a minimal quorum, removing its complement (which is a maximal allowed faulty set according to FBA) from the graph, and then determining how many more nodes, at minimum, must be removed to disconnect the graph (Figure 9). Moreover, instead of just removing the maximal faulty sets, we also simulate how maliciously changing their thresholds can affect the overlay computed by Constellation (Figure 10). For lack of space, the results appear in Appendix G. They show that constellation is robust in those situations.

6.2 Testbed Evaluation

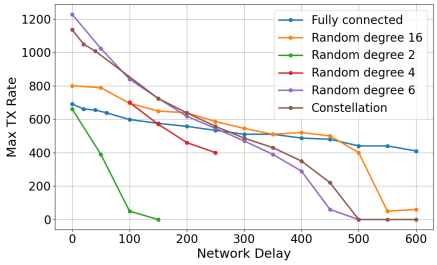
To evaluate Constellation in a more realistic setting, we use a testbed on Amazon EC2 to perform two sets of experiments using stellar-core [2], Stellar's production implementation of the Stellar blockchain.

1. First, we conduct a top-tier only evaluation in which we evaluate the Constellation overlay against random topologies in an FBA system that is a replica of the top-tier of the Stellar network, as it was in October 2024, and without second-tier nodes. This consists of 23 nodes belonging to 7 organization, each running 3 nodes except one running 5 nodes, and all with threshold 5 out of a universe consisting of the 7 organizations.
2. Second, we conduct an evaluation with both top-tier and second-tier nodes, pitching Constellation against random topologies in a replica of the Stellar network, as it was in October 2024; however, because we did not have sufficient hardware to run all nodes on the testbed, we only kept the top 256 nodes that are most connected in the overlay. In this FBAS, the top-tier is the same as above.

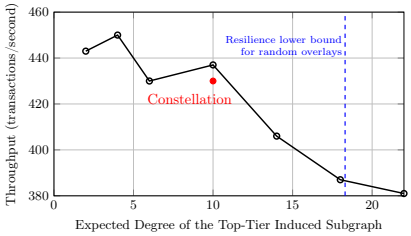
The testbed consists of 6 Amazon EC2 instances: 3 m5a.24xlarge and 3 m5a.12xlarge instances, in the same geographical zone. Each of the Amazon EC2 instances runs roughly 1/6th of the stellar-core [2] validators in the FBA system (with 256 validators, this is less than 1 validator per cpu even for the m5a.12xlarge instances, which have 48 cpus). The stellar-core validators are connected through an IP-layer network and the whole system is managed and deployed using Kubernetes. We estimate that the inter-validators latency is lower than 10 milliseconds, and, on top of this, the testbed allows us to add a configurable, constant simulated latency on all inter-validator communication.

To evaluate a given overlay topology, we fix the overlay connections of the stellar-core validators to obtain exactly the given topology. Then, using a workload of random token-transfer transactions, we inject transactions at several points in the network at a fixed total throughput (i.e. a fixed total number of transactions per second). To find the maximum throughput achievable, we perform a binary search, starting with a conservative upper bound, to find the maximum throughput that the system is a able to sustain for 15 minutes without degradation of key performance metrics.

In the first experiment, we compare the performance of the Constellation overlay and random k-regular overlays for $k = 2, 4, 6, 12, 16$, and 22 (the complete graph), varying additional network latency added to all links. The results appear in Figure 3a. We see that, under typical worldwide inter-datacenter latency (less than 150ms), Constellation outperforms all but the random graph of degree 6. Moreover, for a fair comparison, we need to consider only the random overlays that are FBA-resilient (Definition 3) with high probability. For this, separate simulations show that we need a degree of at least 18, and Constellation clearly outperforms even the degree-16 random overlay (the closest overlay in our benchmark), by up to 50% at 0 added latency.



(a) Latency vs Throughput in random overlays and in the Constellation overlay in a 7-organization FBAS



(b) Throughput in 256-nodes random overlays with varying top-tier degree

Fig. 3: Maximal transaction rate achievable running stellar-core on the testbed.

In the second experiment, we first generate an Erdős–Rényi graph of expected degree 10 spanning the whole 256 nodes. Then, we obtain our test overlays by adding another Erdős–Rényi graph produced among the top-tier only, varying the expected degree of this second graph from 2 to 22. Finally, we also produce a last overlay by adding the Constellation top-tier overlay to the initial degree-10 random graph. The results appear in Figure 3b and show that Constellation has a non-negligible advantage in more realistic scenarios too, where there are a large number of second-tier nodes. Indeed, as previously, with a random overlay, satisfying the FBAS failure assumptions among the top-tier with high probability requires an expected degree of more than 18 (the blue line in the figure), which corresponds to a throughput of approximately 385 transactions per second, while with Constellation we achieve roughly 430 transactions per second; this is an advantage of about 11%.

7 Related Work

Classic structured p2p overlays such as Chord, CAN, Tapestry, Kademlia, and Pastry have been fundamental in shaping the landscape of distributed systems [41,39,32,48,37]. In particular, Kademila is widely adopted by the blockchain community. E.g., Ethereum uses Kademila for address discovery [1], while the Interplanetary File System (IPFS) uses it for data discovery and routing [42]. KadCast proposes a Kademlia-based p2p overlay for faster message dissemination in blockchains [38]. Kadabra builds upon Kademlia and lets nodes improve their peer configurations by learning from and adapting to the environment [46].

While Ethereum uses Kademlia for peer discovery, consensus nodes disseminate data using gossipsub, which uses a random mesh [5,6]. Bitcoin also uses a random overlay [21]. Recent works have proposed newer unstructured p2p overlay designs for blockchains, particularly to account for the heterogeneity of nodes. Perigee [31] proposes an adaptive topology-updating algorithm that uses

an exploration-exploitation framework to discover an efficient topology. Zarin et al. [45] propose an overlay design in which nodes are segmented into distinct domains depending on their type and resource availability, while using Kademlia for node discovery. All of these works ignore inherent consensus-level relationships, such as FBA's agreement requirements, that might exist between nodes.

There are few works that build a Byzantine-resilient overlay which is secure under the same assumptions as used by the consensus protocol. Notable exceptions are the recent works by Coretti et al. [13] and Liu-Zhang et al [26,26], which propose random graph overlays with the degree of each node proportional to the amount of stake it has. Cohen et al [12] and Tsimos et al. [43] propose efficient Byzantine-resilient gossip protocols for PKI settings.

Networking in FBA systems has been sporadically studied. Vytautas Tumas et al. [44] investigate network-level attacks in the XRP Ledger, which uses a model different from the FBA model but where nodes can also make subjective trust assumptions. The study introduces a novel robustness metric to assess resilience to node failures, revealing vulnerabilities in the XRP Ledger Consensus Protocol and proposing a mitigation strategy to enhance robustness. André Gaul et al. [19] focus on assessing the level of decentralization of FBA networks using centrality measures. They explore three approaches for obtaining centrality measures for FBAS nodes, including adaptations of established graph and hypergraph-based measures, along with a newly developed measure based on node intactness—a crucial aspect of the FBAS model. Florian et al. [16] study the dynamics of reconfiguration in FBA systems.

Sophisticated view-maintenance and view-sampling algorithms [8,36,35,47,7] can, under Sybil-resilience assumptions, ensure resilience to Byzantine, network-level attacks such as eclipse attacks [21], and such algorithms could be considered to connect second-tier nodes in a FBA system.

8 Conclusion

We have presented and evaluated Constellation, an algorithm to construct overlays for Federated Byzantine Agreement (FBA) systems that are both performant in the normal case and resilient to failures, and a practical scheme to deploy the Constellation algorithm in an FBA system such as the Stellar network.

For symmetric FBA systems, we have shown that Constellation guarantees connectivity of the overlay under all failures that must be tolerated according to the FBA model, and that it achieves an average degree close to optimal. In the general case, we have shown that Constellation overlays remain of diameter 3 under a large fraction of failures. Finally, performance metrics collected both in simulation and in a system deployed on live testbed shows that Constellation achieves better node degree and better measured transaction throughput, under realistic latencies, that random overlays.

In future work, it would be interesting to investigate probabilistic solutions which guarantee resilience only probabilistically but achieve much lower degree (with perhaps higher diameter), and also to investigate more decentralized solutions which do rely on using the blockchain to agree on the FBA configuration.

References

1. Ethereum Peer Discovery Protocol (2024), https://github.com/ethereum/devp2p/blob/master/discv5/discv5.md
2. The Stellar-Core Github Repository (2024), https://github.com/stellar/stellar-core/
3. Website of the Stellar Development Foundation (2024), https://stellar.org/
4. Constellation Pre-Proceedings Version (2025), https://zenodo.org/records/14873487
5. Ethereum Consensus Specs (2025), https://ethereum.github.io/consensus-specs/
6. Gossipsub (2025), https://github.com/libp2p/specs/tree/master/pubsub/gossipsub
7. Auvolat, A., Bromberg, Y.D., Frey, D., Mvondo, D., Taïani, F.: Basalt: A Rock-Solid Byzantine-Tolerant Peer Sampling for Very Large Decentralized Networks. In: Proceedings of the 24th International Middleware Conference. pp. 111–123. Middleware '23, Association for Computing Machinery, New York, NY, USA (Nov 2023). https://doi.org/10.1145/3590140.3629109
8. Bortnikov, E., Gurevich, M., Keidar, I., Kliot, G., Shraer, A.: Brahms: Byzantine resilient random membership sampling. Comput. Netw. **53**(13), 2340–2359 (Aug 2009). https://doi.org/10.1016/j.comnet.2009.03.008
9. Cachin, C.: Asymmetric Distributed Trust. In: International Conference on Distributed Computing and Networking 2021 (2021)
10. Cachin, C., Losa, G., Zanolini, L.: Quorum Systems in Permissionless Network (Nov 2022). https://doi.org/10.48550/arXiv.2211.05630
11. Cachin, C., Zanolini, L.: Asymmetric Asynchronous Byzantine Consensus. In: Garcia-Alfaro, J., Muñoz-Tapia, J.L., Navarro-Arribas, G., Soriano, M. (eds.) Data Privacy Management, Cryptocurrencies and Blockchain Technology. pp. 192–207. Lecture Notes in Computer Science, Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-030-93944-1_13
12. Cohen, R., Loss, J., Moran, T.: Efficient Agreement Over Byzantine Gossip (2023)
13. Coretti, S., Kiayias, A., Moore, C., Russell, A.: The Generals' Scuttlebutt: Byzantine-Resilient Gossip Protocols. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 595–608 (2022)
14. Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J.: Epidemic Algorithms for Replicated Database Maintenance. In: Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing - PODC '87. pp. 1–12. ACM Press, Vancouver, British Columbia, Canada (1987). https://doi.org/10.1145/41840.41841
15. Douceur, J.R.: The sybil attack. In: International Workshop on Peer-to-Peer Systems. pp. 251–260. Springer (2002)
16. Florian, M., Henningsen, S., Ndolo, C., Scheuermann, B.: The sum of its parts: Analysis of federated byzantine agreement systems. Distributed Computing **35**(5), 399–417 (Oct 2022). https://doi.org/10.1007/s00446-022-00430-0
17. García-Pérez, Á., Gotsman, A.: Federated Byzantine Quorum Systems. In: 22nd International Conference on Principles of Distributed Systems (OPODIS 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2018)
18. García-Pérez, Á., Schett, M.A.: Deconstructing Stellar Consensus. In: DROPS-IDN/v2/Document/10.4230/LIPIcs.OPODIS.2019.5. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPIcs.OPODIS.2019.5

19. Gaul, A., Liesen, J.: Centrality of Nodes in Federated Byzantine Agreement Systems (2020)
20. Govorčin, J., Škrekovski, R.: On the Connectivity of Cartesian Product of Graphs. Ars mathematica contemporanea **7**(2) (2013)
21. Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse Attacks on Bitcoin's Peer-to-Peer Network. In: 24th {USENIX} Security Symposium ({USENIX} Security 15). pp. 129–144 (2015)
22. Karp, R., Schindelhauer, C., Shenker, S., Vocking, B.: Randomized Rumor Spreading. In: Proceedings 41st Annual Symposium on Foundations of Computer Science. pp. 565–574. IEEE (2000)
23. Knuth, D.E.: The Art of Computer Programming: Combinatorial Algorithms, Part 1. Addison-Wesley Professional, 1st edn. (2011)
24. Li, X., Chan, E., Lesani, M.: Quorum Subsumption for Heterogeneous Quorum Systems. In: DROPS-IDN/v2/Document/10.4230/LIPIcs.DISC.2023.28. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2023). https://doi.org/10.4230/LIPIcs.DISC.2023.28
25. Liouville, B.: Sur la connectivité des produits de graphes. (1978)
26. Liu-Zhang, C.D., Matt, C., Maurer, U., Rito, G., Thomsen, S.E.: Practical Provably Secure Flooding for Blockchains. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology – ASIACRYPT 2022. pp. 774–805. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-22963-3_26
27. Liu-Zhang, C.D., Matt, C., Thomsen, S.E.: Asymptotically Optimal Message Dissemination with Applications to Blockchains. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024. pp. 64–95. Springer Nature Switzerland, Cham (2024). https://doi.org/10.1007/978-3-031-58734-4_3
28. 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. pp. 80–96. SOSP '19, Association for Computing Machinery, Huntsville, Ontario, Canada (Oct 2019). https://doi.org/10.1145/3341301.3359636
29. Losa, G., Gafni, E., Mazières, D.: Stellar Consensus by Instantiation. In: Suomela, J. (ed.) 33rd International Symposium on Distributed Computing (DISC 2019). Leibniz International Proceedings in Informatics (LIPIcs), vol. 146, pp. 27:1–27:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2019). https://doi.org/10.4230/LIPIcs.DISC.2019.27
30. Lü, M., Wu, C., Chen, G.L., Lv, C.: On Super Connectivity of Cartesian Product Graphs. Networks: An International Journal **52**(2), 78–87 (2008)
31. Mao, Y., Deb, S., Venkatakrishnan, S.B., Kannan, S., Srinivasan, K.: Perigee: Efficient Peer-to-Peer Network Design for Blockchains. In: Proceedings of the 39th Symposium on Principles of Distributed Computing. pp. 428–437 (2020)
32. Maymounkov, P., Mazieres, D.: Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In: International Workshop on Peer-to-Peer Systems. pp. 53–65. Springer (2002)
33. Mazieres, D.: The Stellar Consensus Protocol: A Federated Model for Internet-Level Consensus. Stellar Development Foundation (2015)
34. Parsonage, E., Nguyen, H.X., Bowden, R., Knight, S., Falkner, N., Roughan, M.: Generalized Graph Products for Network Design and Analysis. In: 2011 19th IEEE International Conference on Network Protocols. pp. 79–88. IEEE (2011)
35. Pigaglio, M., Bruneau-Queyreix, J., Bromberg, Y.D., Frey, D., Rivière, E., Réveillère, L.: RAPTEE: Leveraging trusted execution environments for Byzantine-tolerant peer sampling services. In: 2022 IEEE 42nd International Conference

on Distributed Computing Systems (ICDCS). pp. 603–613 (Jul 2022). https://doi.org/10.1109/ICDCS54860.2022.00064
36. Pilet, A.B., Frey, D., Taïani, F.: Foiling Sybils with HAPS in Permissionless Systems: An Address-based Peer Sampling Service. In: 2020 IEEE Symposium on Computers and Communications (ISCC). pp. 1–6 (Jul 2020). https://doi.org/10.1109/ISCC50000.2020.9219606
37. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A Scalable Content-Addressable Network. In: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications. pp. 161–172 (2001)
38. Rohrer, E., Tschorsch, F.: Kadcast: A Structured Approach to Broadcast in Blockchain Networks. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies. pp. 199–213. AFT '19, Association for Computing Machinery, New York, NY, USA (Oct 2019). https://doi.org/10.1145/3318041.3355469
39. Rowstron, A., Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In: Middleware 2001: IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, Germany, November 12–16, 2001 Proceedings 2. pp. 329–350. Springer (2001)
40. Špacapan, S.: Connectivity of Cartesian Products of Graphs. Applied Mathematics Letters **21**(7), 682–685 (2008)
41. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. IEEE/ACM Transactions on networking **11**(1), 17–32 (2003)
42. Trautwein, D., Raman, A., Tyson, G., Castro, I., Scott, W., Schubotz, M., Gipp, B., Psaras, Y.: Design and Evaluation of IPFS: A Storage Layer for the Decentralized Web. In: Proceedings of the ACM SIGCOMM 2022 Conference. pp. 739–752 (2022)
43. Tsimos, G., Loss, J., Papamanthou, C.: Gossiping for Communication-Efficient Broadcast. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology – CRYPTO 2022. pp. 439–469. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-15982-4_15
44. Tumas, V., Rivera, S., Magoni, D., State, R.: Federated Byzantine Agreement Protocol Robustness to Targeted Network Attacks. In: 2023 IEEE Symposium on Computers and Communications (ISCC). pp. 443–449. IEEE Computer Society, Los Alamitos, CA, USA (jul 2023). https://doi.org/10.1109/ISCC58397.2023.10217935, https://doi.ieeecomputersociety.org/10.1109/ISCC58397.2023.10217935
45. Zarin, N., Sheff, I., Roos, S.: Blockchain Nodes are Heterogeneous and Your P2P Overlay Should be Too: PODS (Jun 2023). https://doi.org/10.48550/arXiv.2306.16153
46. Zhang, Y., Bojja Venkatakrishnan, S.: Kadabra: Adapting Kademlia for the Decentralized Web. In: International Conference on Financial Cryptography and Data Security. pp. 327–345. Springer (2023)
47. Zhang, Y., Bojja Venkatakrishnan, S.: Honeybee: Decentralized Peer Sampling with Verifiable Random Walks for Blockchain Data Sharding. arXiv preprint arXiv:2402.16201 (2024)
48. Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D., Kubiatowicz, J.D.: Tapestry: A Resilient Global-Scale Overlay for Service Deployment. IEEE Journal on selected areas in communications **22**(1), 41–53 (2004)

(a) Each node establishes 2 random outgoing connections

(b) A better topology

Fig. 4: A random overlay and an optimized overlay on 5 nodes.

A Motivating Examples

To motivate the design of Constellation, we start by analyzing three concrete examples. Each time, we contrast a simple strategy where nodes establish a fixed number of random connections, without coordination, with an ad-hoc topology that we design using global knowledge of the FBAS configuration (universe and threshold) of each node. We will see that the random approach incurs significantly higher node degree in the overlay.

A.1 A Simple 5-Node Network

Consider a network with 5 organizations $\mathcal{O} = \{O_i | i \in 1..5\}$, each running a single node n_i with the universe $\mathcal{O}$ and threshold 4 (i.e. all nodes assume that at least 4 out of the 5 nodes will be available and well-behaved). In this configuration, a resilient overlay must ensure connectivity despite the failure of any one node.

A simple strategy consists in having each node randomly pick 2 other nodes to connect to in order to form the overlay. This will result in a resilient overlay with high probability. Figure 4a represents a possible result. The edge's direction indicate which node chose to establish the connection, and we can verify that each node has two outgoing edges. The resulting overlay is the undirected graph obtained by making the edges undirected and merging double edges.

In this case, we get average degree of 3.2 and max degree of 4. However, we can do much better, as in Figure 4b, where removing any one node still leaves the graph connected, all the nodes only have degree of 2, and the diameter is still 2. In contrast, in Figure 4a, nodes end up with a degree higher than needed because, without coordination, there is a very high chance of establishing redundant connections.

In Appendix A.3, we present an example of a topology for the top-tier nodes in today's Stellar network that achieves a significantly better degree compared to random. We conclude that having each node randomly establish a fixed number of connection is wasteful because, without coordination, there is a high chance that a significant number of redundant connections will be created. In contrast,

Constellation uses global knowledge of the FBAS configurations of the nodes to coordinate building a resilient and efficient overlay.

A.2 A Generic Approach for Symmetric Networks

In this motivating example, we consider a system of $|\mathcal{O}| = 3k + 1$ organizations each running 3 nodes (so $|N| = 9k + 3$) and where all nodes have the same universe $\mathcal{O}$ and threshold $2k + 1$. Note that, assuming that all nodes of an organization fail maliciously or none do, this mimics a traditional 2/3-threshold Byzantine quorum system.

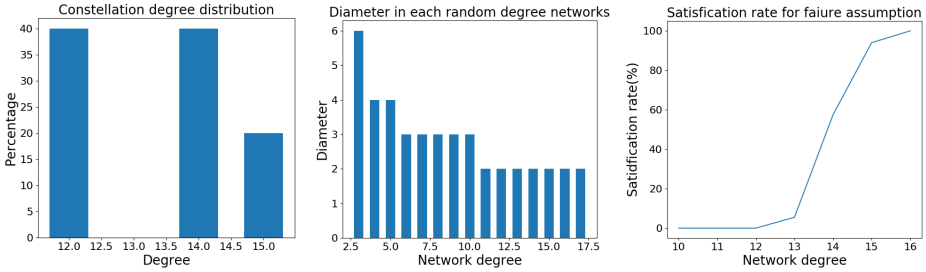
Also note that, if a node n is not connected in the overlay to at least 2 nodes of each of $k + 1$ organizations, then clearly we can crash enough nodes to disconnect n from all other nodes and still satisfy n's failure assumptions. Thus, it is not possible to build a resilient overlay where any node has a degree smaller than $2(k + 1)$.

The construction we now present is a special case of the full Constellation algorithm presented in Sections 4 and F. To build an overlay for this system, we first create what we call a hypergraph over the organizations. For this, we partition the organizations into 2 clusters of size k ($\{O_1^1, O_2^1, \ldots, O_k^1\}$ and $\{O_1^2, O_2^2, \ldots, O_k^2\}$) and one cluster of size $k + 1$ ($\{O_1^3, O_2^3, \ldots, O_{k+1}^3\}$). Then, we create a complete graph inside each cluster (i.e. creating edges $\langle O_m^i, O_n^i \rangle$ for every $m \neq n$). Finally, we create inter-cluster connections by adding an edge $\langle O_m^i, O_n^j \rangle$ if and only if $m \equiv n \pmod{k}$. An example for $k = 3$ appears in Figure 1a. Note that this creates a perfect matching between clusters 1 and 2 and, because of the difference in cardinality, almost a perfect matching between clusters 1 and 3 and clusters 2 and 3. Also note that the resulting graph is close to the product of the complete graph on k vertices and the triangle graph.

Now that we have a graph over organizations, for every two organizations that form an edge $\langle A, B \rangle$, we connect their nodes using one of two different patterns. If both A and B belong to the same cluster, we create a bipartite graph where each node of one organization is connected to exactly two nodes of the other organization, as shown in Figure 1b. If A and B are in different clusters, we connect each node of A to every node of B, obtaining a complete bipartite graph as shown in Figure 1c. Additionally, we connect all nodes of an organization with each other.

Note we could have used complete bipartite graphs for all inter-organization edges and obtained a diameter of 2. However, using the more parsimonious pattern of Figure 1b for intra-cluster edges allows us to achieve a smaller degree while keeping the diameter at 2. Also note that we achieve a maximum node degree of $2k+9$, which is close to the lower bound of $2(k+1)$ dictated by the node's FBA configuration. For example, in Figure 1a, nodes belonging to organization O_1^1 and O_1^2 have degree $2k + 9 = 15$.

We now compare the overlay that we obtain using the generic construction and with the random strategy for 10 organizations and 3 nodes per organization. The results appear in Figure 5. Figure 5a represents the distribution of node degree obtained with the construction of Figure 1. We see that 40% of the nodes



(a) Constellation degree distribution

(b) Average diameter in random regular graphs, depending on node degree

(c) Percentage of nodes whose failure assumptions are satisfied in random regular graphs, depending on node degree

Fig. 5: Comparison between Constellation and random in 10-org network.

have a degree of 12, 40% have a degree of 14, and the remaining 20% have a degree of 15.

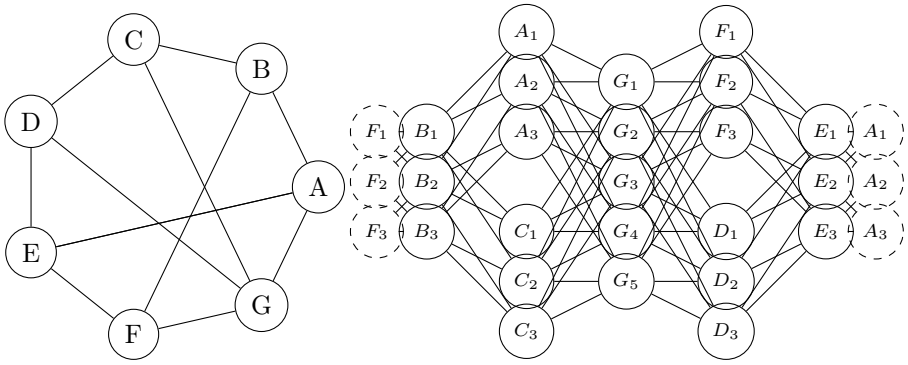
In Figures 5b and 5c, we generate random overlays by having each node pick a fixed number of random neighbors and plot the average node degree obtained and the percentage of nodes whose failure assumptions are satisfied. In Figure 5b, we can see that to achieve a diameter of 2, like in the construction of Figure 1a, we need to reach an average node degree of 11. In Figure 5c, we can see that only at a degree of 16 do we obtain a resilient overlay (where the overlay remains connected even under failures as long as the failure assumptions of the remaining nodes are satisfied). With Constellation, we meet both criteria (diameter of 2 and resilience) with a maximum degree of 15.

A.3 Example of an Efficient Topology for the Stellar Network

The Stellar network currently comprises seven top-tier organizations $\mathcal{O} = \{A, B, C, D, E, F, G\}$, with six organizations $\{A, B, C, D, E, F\}$ operating 3 nodes and one organization G operating 5 nodes, for a total of 23 nodes[13]. All nodes are configured with a threshold of 5 out of all the 7 organizations.

Nodes in the Stellar network currently establish random connections to a fixed number of other nodes. To experiment with this strategy while ensuring resilience and a diameter of at most 2, we run a series of 10 experiments in which, each time, starting with $k = 1$ we create a graph where each node picks k random neighbors, and we increment the value of k until we obtain a resilient overlay with diameter at most 2. On average over those 10 experiments, we obtain a final degree of $k = 16$.

[13] The structure of the Stellar network can be explored at `https://stellarbeat.io` (accessed February 12th, 2025)



(a) First step: organization-level connections

(b) Second step: realizing inter-organization connections at the node level. Dotted nodes are repeated for clarity of the graph layout. Note that intra-organization edges are not represented, but the nodes of each organization form a complete graph.

Fig. 6: A resilient and efficient topology on Stellar's current 7-organization network

In contrast, consider the overlay of Figure 6. We design this overlay in two steps. We start in Figure 6a by creating what we call a hypergraph connecting organizations (which can be though of as sets of nodes), minimizing degree but making sure that each organization has edges to at least 3 other organizations. Note that, since we know that node have a threshold of 5 out of all the 7 organizations, connecting to $7 - 5 + 1 = 3$ organizations ensures connecting to at least one available and well-behaved organization. The best we can do to minimize degree is to use a graph as in Figure 6a where only one vertex has degree 4 (this is vertex G) while all the others have degree 3.

Next, we realize the inter-organization connections at the node level. To do so, for each edge connecting two organizations in Figure 6a, we create inter-node connections that minimize node degree but ensure that, for each edge between two organizations, each node of one organization is connected to more than half of the nodes of the other organization.

Finally, we connect every pair of nodes that belong to the same organization. The resulting overlay appears in Figure 6b. Note that we achieve a diameter of 2. Moreover, in contrast to the average degree of 16 in the random case, the maximum node degree is now 11, a significant improvement. It is also easy to see that the overlay remains connected even if up to two organizations (plus a minority of the nodes of each remaining organization) fail, which makes the topology FBA-resilient.

As in the example of §A.1, this shows that the random strategy can be much improved upon, this time with the configurations observed in the Stellar network today.

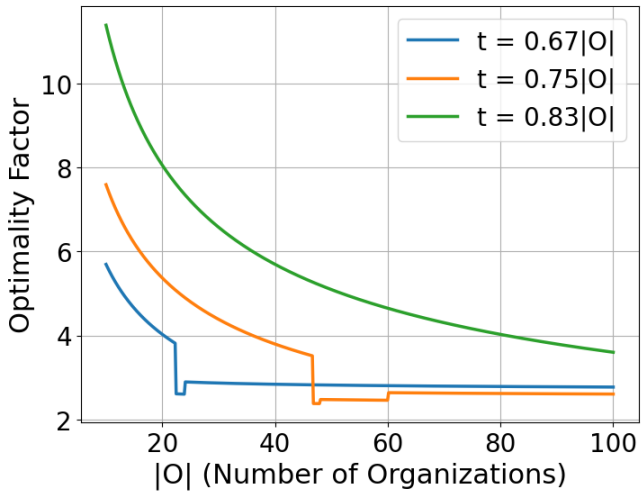


Fig. 7: Optimality factor as a function of the number of organizations, for a few values of t.

The Constellation algorithm that we present in Section 4 generalizes what we have done in this section in order to compute an optimized overlay for any given regular FBAS.

B Proof of Proposition 1

Proof. Consider any two nodes $u, v \in N$. If u and v belong to the same organization O, then by construction u has an edge to v. If u and v are in the different organizations, O_u and O_v respectively, and O_u and O_v are in the same cluster, then by construction u has an edge to a node in O_v which in turn has an edge to v. Hence, the shortest path length between u and v is also at most 2 in this case. If O_u and O_v are in different clusters and $(O_u, O_v) \in H$ then O_u and O_v form a complete bipartite graph in G. Therefore, $(u, v) \in E$ and the length of the shortest path between u and v is 1.

If O_u and O_v are in different clusters but $(O_u, O_v) \notin H$, then there must be an organization O' that belongs to O_u's cluster with $(O', O_v) \in H$. For any node in O' that connects to u, it must also connect with all the nodes in O_v including v, so that the shortest path between u and v is no more than 2 hops in this case as well. .

C Proof of Theorem 1

In this section, we present a proof of Theorem 1 and, in Figure 7 a graphical depiction of the bound proved for a few threshold values.

Theorem 1. *In the symmetric case with $t_v = t, \mathcal{T}_v = \mathcal{O}$ for all $v \in N$, each cluster having the same number of organizations, and each organization having the same number of validators, the validator degree achieved by Constellation exceeds the optimal degree by a factor at most $2 + \frac{2}{k^2 f} + \frac{1}{\sqrt{|\mathcal{O}|}-1/2}$ for $\frac{2|\mathcal{O}|}{3} \leq t < |\mathcal{O}| + 2 - 2\sqrt{|\mathcal{O}|}$ and by a factor at most $\frac{6}{\sqrt{|\mathcal{O}|f}}$ for $|\mathcal{O}| + 2 - 2\sqrt{|\mathcal{O}|} \leq t \leq |\mathcal{O}|$, where $f = \frac{|\mathcal{O}|-t}{|\mathcal{O}|}$ denotes the maximum fraction of organizations that can fail. (Proof in Appendix C)*

Proof. Let OPT denote the validator degree achieved by an optimal overlay construction algorithm. We have seen that an organization must form hyperedges with at least $|\mathcal{O}| - t + 1$ other organizations to guarantee connectivity to at least one available organization. We have also seen that a validator in the organization must form edges to at least half of the validators of any other organization it seeks to form hyperedges with. These two observations imply a lower bound of

$$d_v \geq OPT \geq (|\mathcal{O}| - t + 1) \left\lceil \frac{N}{2|\mathcal{O}|} \right\rceil \tag{3}$$

on the degree d_v for any validator $v \in N$. Here $N/|\mathcal{O}|$ is the number of validators in each organization.

To compare the lower bound in Equation (3) with the degree achieved in Constellation, first let us consider the case where $\frac{2}{3}|\mathcal{O}| \leq t < |\mathcal{O}| + 2 - 2\sqrt{|\mathcal{O}|}$. Let

$$k^* = \frac{2 + |\mathcal{O}| - t}{2} - \frac{\sqrt{4 + |\mathcal{O}|^2 - 4t - 2|\mathcal{O}|t + t^2}}{2} \tag{4}$$

and let $k = \lfloor k^* \rfloor$ be the value that achieves equality in Equation (2). Let $\epsilon = k^* - k$ Clearly, $0 \leq \epsilon < 1$. Therefore, the achieved number of hyperedges with other organizations can be bounded as

$$\begin{aligned} k - 1 + \frac{|\mathcal{O}|}{k} - 1 &= k^* - \epsilon - 1 + \frac{|\mathcal{O}|}{k^* - \epsilon} - 1 \\ &= k^* - \epsilon - 1 + \frac{|\mathcal{O}|}{k^*}\left(1 + \frac{\epsilon}{k^*} + \frac{\epsilon^2}{(k^*)^2} + \ldots\right) - 1 \\ &= |\mathcal{O}| - t - \epsilon + \frac{|\mathcal{O}|\epsilon}{k^* k^* (1 - \epsilon/k^*)} \\ &\leq |\mathcal{O}| - t + \frac{|\mathcal{O}|}{k^* k} \leq |\mathcal{O}| - t + \frac{|\mathcal{O}|}{k^2}. \end{aligned} \tag{5}$$

Since a validator of an organization makes at most $N/|\mathcal{O}|$ edges to validators of any other organization it has an hyperedge with, and all validators within an organization have edges to each other, the total achieved degree d_v of a validator in Constellation is at most

$$d_v \leq \left(|\mathcal{O}| - t + \frac{|\mathcal{O}|}{k^2}\right)\left(\frac{N}{|\mathcal{O}|}\right) + \left(\frac{N}{|\mathcal{O}|} - 1\right). \tag{6}$$

Therefore,

$$\begin{aligned}
\frac{d_v}{OPT} &\leq \frac{\left(|\mathcal{O}| - t + \frac{|\mathcal{O}|}{k^2}\right)\left(\frac{N}{|\mathcal{O}|}\right) + \left(\frac{N}{|\mathcal{O}|} - 1\right)}{(|\mathcal{O}| - t + 1)\frac{N}{2|\mathcal{O}|}} \\
&\leq \left(1 + \frac{|\mathcal{O}| - k^2}{k^2(|\mathcal{O}| - t + 1)}\right) 2 + \frac{\frac{N}{|\mathcal{O}|} - 1}{(|\mathcal{O}| - t + 1)\frac{N}{2|\mathcal{O}|}} \\
&\leq 2 + \frac{2|\mathcal{O}| - 2k^2}{k^2(|\mathcal{O}| - t + 1)} + \frac{2(N - |\mathcal{O}|)}{(|\mathcal{O}| - t + 1)N} \\
&\leq 2 + \frac{2|\mathcal{O}|}{k^2(f|\mathcal{O}|)} + \frac{2(1 - \frac{|\mathcal{O}|}{N})}{(2\sqrt{|\mathcal{O}|} - 1)} \\
&\leq 2 + \frac{2}{k^2 f} + \frac{1 - \frac{|\mathcal{O}|}{N}}{\sqrt{|\mathcal{O}|} - 1/2} \\
&\leq 2 + \frac{2}{k^2 f} + \frac{1}{\sqrt{|\mathcal{O}|} - 1/2}. \qquad (7)
\end{aligned}$$

Next, consider the case where $|\mathcal{O}| + 2 - 2\sqrt{|\mathcal{O}|} \leq t \leq |\mathcal{O}|$. In this case, we use $k = \lfloor \sqrt{|\mathcal{O}|} \rfloor$ clusters yielding a total number of hyperedges for an organization of

$$\begin{aligned}
\lfloor \sqrt{|\mathcal{O}|} \rfloor - 1 + \frac{|\mathcal{O}|}{\lfloor \sqrt{|\mathcal{O}|} \rfloor} - 1 &\leq \sqrt{|\mathcal{O}|} + \frac{|\mathcal{O}|}{\sqrt{|\mathcal{O}|} - 1} - 2 \\
&\leq \sqrt{|\mathcal{O}|} + \sqrt{|\mathcal{O}|}\frac{\sqrt{|\mathcal{O}|}}{\sqrt{|\mathcal{O}|} - 1} - 2 \\
&\leq 3\sqrt{|\mathcal{O}|} - 2. \qquad (8)
\end{aligned}$$

The total achieved degree of a Constellation validator is therefore

$$d_v \leq (3\sqrt{|\mathcal{O}|} - 2)\frac{N}{|\mathcal{O}|} + \left(\frac{N}{|\mathcal{O}|} - 1\right). \qquad (9)$$

The optimum degree achievable in this case is

$$OPT \geq (|\mathcal{O}| - t + 1)\left\lceil \frac{N}{2|\mathcal{O}|} \right\rceil \geq f|\mathcal{O}|\frac{N}{2|\mathcal{O}|} = \frac{fN}{2}. \qquad (10)$$

Therefore,

$$\frac{d_v}{OPT} \leq \frac{6}{\sqrt{|\mathcal{O}|}f} - \frac{2}{|\mathcal{O}|f} - \frac{2}{fN} \leq \frac{6}{\sqrt{|\mathcal{O}|}f}. \qquad (11)$$

D Proof of Theorem 2

Theorem 2. *For symmetric thresholds $t_v = t, \mathcal{T}_v = \mathcal{O}$ for all $v \in N$, $t \geq 2|\mathcal{O}|/3$, each cluster having the same number of organizations, and each organization*

having the same number of validators, the Consthe number of organizations that must be removed to disconnect the Constellation network is at least $|\mathcal{O}| - t + 1$. *(Proof in Appendix D)*

Proof. In the hypergraph H, consider any subset of organizations $S \subseteq \mathcal{O}$ that is connected. Here connected means, for any organizations $O_1, O_2 \in S$ either $(O_1, O_2) \in H$ or there exists a sequence of hyperedges $(O_1, O_1'), (O_1', O_2'), \ldots, (O_{l-1}', O_l'), (O_l', O_2)$ for $l \geq 1$ with $O_1', O_2', \ldots, O_l' \in S$ and $(O_1, O_1'), (O_1', O_2'), \ldots, (O_{l-1}', O_l'), (O_l', O_2) \in H$. If S is a connected subset of organizations in H, then the set of validators $N_S = \{v \in N : v \in O, O \in S\}$ form a connected subgraph in G. To see this, consider any two validators $u, v \in N_S$. Say, $u \in O_1$ and $v \in O_2$. If $O_1 = O_2$, u and v are clearly connected since all validators within an organization have edges to each other. If $O_1 \neq O_2$, since S forms a connected subgraph, either $(O_1, O_2) \in H$ or there exists a sequence of hyperedges $(O_1, O_1'), (O_1', O_2'), \ldots, (O_{l-1}', O_l'), (O_l', O_2)$ for $l \geq 1$ with $O_1', O_2', \ldots, O_l' \in S$ and $(O_1, O_1'), (O_1', O_2'), \ldots, (O_{l-1}', O_l'), (O_l', O_2) \in H$. If $(O_1, O_2) \in H$, then u has an edge to at least one available node in O_2 through which it is connected to v. If $(O_1, O_2) \notin H$ but there exists a path $(O_1, O_1'), (O_1', O_2'), \ldots, (O_{l-1}', O_l'), (O_l', O_2)$ from O_1 to O_2 in S (with the hyperedges belonging to H), then u is connected to at least one available node $o_1' \in O_1'$. Similarly, o_1' is connected with at least one available node $o_2' \in O_2$, and so on. The penultimate organization has validator o_l' that has an edge to an available node in O_2, which in turn is connected to v. Thus, there exists a path from u to v proving that N_S forms a connected subgraph. To analyze the connectivity of Constellation's topology, it therefore suffices to analyze the connectivity within Constellation's hypergraph.

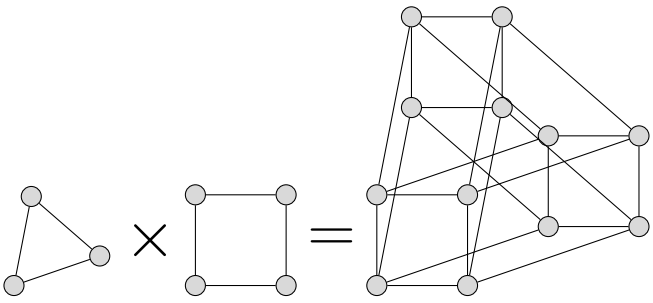


Fig. 8: Example of product of the two graphs K_3 and K_4.

In the symmetric case when clusters have the same number of organizations, the hypergraph created by Constellation has a structure that resembles the Cartesian product of the complete graphs K_k and $K_{|\mathcal{O}|/k}$, where K_i denotes the complete graph with i vertices (See Figure 8 for $k = 3$ and $|\mathcal{O}| = 12$)). For this product graph, the connectivity number (i.e., minimum number of vertices needed to disconnect the graph, or form a trivial graph with just a single vertex)

is given by Liouville's formula [20]:

$$\kappa(K_k \square K_{|\mathcal{O}|/k}) = \min\{\kappa(K_k)|K_{|\mathcal{O}|/k}|, \kappa(K_{|\mathcal{O}|/k})|K_k|, \delta(K_k) + \delta(K_{|\mathcal{O}|/k})\} \quad (12)$$

where $\square$ denotes cartesian product of graphs and $\delta(G)$ is the minimum degree of graph G. For a complete graph K_i, the connectivity number $\kappa(K_i) = i - 1$ and $\delta(K_i) = i - 1$. Therefore,

$$\kappa(K_k \square K_{|\mathcal{O}|/k}) = \min\left\{(k-1)\frac{|\mathcal{O}|}{k}, \left(\frac{|\mathcal{O}|}{k} - 1\right)k, k - 1 + \frac{|\mathcal{O}|}{k} - 1\right\}. \quad (13)$$

However, if each cluster contains at least two organizations, we have

$$\begin{aligned}
&(k-1)\frac{|\mathcal{O}|}{k} \geq k - 1 + \frac{|\mathcal{O}|}{k} - 1 \\
\iff\ & k|\mathcal{O}| - |\mathcal{O}| \geq k^2 - k + |\mathcal{O}| - k \\
\iff\ & k|\mathcal{O}| \geq k^2 - 2k + 2|\mathcal{O}| \\
\iff\ & |\mathcal{O}|(k-2) \geq k(k-2) \\
\iff\ & |\mathcal{O}| \geq k, \qquad (14)
\end{aligned}$$

and

$$\begin{aligned}
&\left(\frac{|\mathcal{O}|}{k} - 1\right)k \geq k - 1 + \frac{|\mathcal{O}|}{k} - 1 \\
\iff\ & |\mathcal{O}| - k \geq k - 1 + |\mathcal{O}|/k - 1 \\
\iff\ & |\mathcal{O}|k - k^2 \geq k^2 - k + |\mathcal{O}| - k \\
\iff\ & |\mathcal{O}|k + 2k \geq 2k^2 + |\mathcal{O}| \\
\iff\ & |\mathcal{O}|(k-1) \geq 2k(k-1) \\
\iff\ & |\mathcal{O}| \geq 2k. \qquad (15)
\end{aligned}$$

Hence,

$$\kappa(K_k \square K_{|\mathcal{O}|/k}) = k - 1 + \frac{|\mathcal{O}|}{k} - 1 \geq |\mathcal{O}| - t + 1, \quad (16)$$

where the last inequality follows from Equation (1). It follows that Constellation is connected if fewer than $|\mathcal{O}| - t + 1$ organizations fail.

E Proof of Proposition 2

Proposition 2. *For k clusters, with each cluster having the same number of organizations, the diameter of the Constellation network is at most 3 even under failure of $|\mathcal{O}|/k + k - 2$ organizations.*
(Proof in Appendix E)

Proof. For any two organizations in the hypergraph H, we show the existence of $|\mathcal{O}|/k + k - 2$ node disjoint paths between them. For $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, |\mathcal{O}|/k$, let $O_{i,j}$ denote the j-th organization in the i-th cluster. Also, for any $i, i' \in \{1, 2, \ldots, k\}$ and $j \in \{1, 2, \ldots, |\mathcal{O}|/k\}$ let $(O_{i,j}, O_{i',j}) \in H$. Since all organizations within a cluster are connected to each other, we have $(O_{i,j}, O_{i,j'} \in H$ for all $i \in \{1, 2, \ldots, k\}$ and $j, j' \in \{1, 2, \ldots, |\mathcal{O}|/k\}$.

Now, consider any two organizations $O_{i,j}$ and $O_{i',j'}$ with $i, i' \in \{1, 2, \ldots, k\}$ and $j, j' \in \{1, 2, \ldots, |\mathcal{O}|/k\}$. If either $i = i'$ or $j = j'$, the organizations have a direct hyperedge between them. Therefore, the failure of any other organization does not impact the connectivity distance between the two organizations. Next, consider the case where $i \neq i'$ and $j \neq j'$, with $i, i' \in \{1, 2, \ldots, k\}$ and $j, j' \in \{1, 2, \ldots, |\mathcal{O}|/k\}$. Consider the following paths:

1. $p_1 = (O_{i,j}, O_{i,j'}), (O_{i,j'}, O_{i',j'})$
2. $p_2 = (O_{i,j}, O_{i',j}), (O_{i',j}, O_{i,j'})$
3. $p_l = (O_{i,j}, O_{i,j''}), (O_{i,j''}, O_{i',j''}), (O_{i',j''}, O_{i,j'})$ for $j'' \in \{1, 2, \ldots, |\mathcal{O}|/k\}, j'' \neq j, j'' \neq j'$. Here $l = 3, 4, \ldots, |\mathcal{O}|/k$ with p_l for each l denoting the path via a distinct j''.
4. $p_l = (O_{i,j}, O_{i'',j}), (O_{i'',j}, O_{i'',j'}), (O_{i'',j'}, O_{i',j'})$ for $i'' \in \{1, 2, \ldots, k\}, i'' \neq i, i'' \neq i'$. Here $l = |\mathcal{O}|/k + 1, \ldots, |\mathcal{O}|/k + k - 2$ with p_l for each l denoting the path via a distinct cluster i''.

Clearly, the paths $p_1, \ldots, p_{|\mathcal{O}|/k+k-2}$ are node disjoint, and have a length of at most 3 hops. Thus, we conclude that the diameter of the Constellation network is at most 3 under failures of up to $|\mathcal{O}|/k + k - 2$ organizations.

F Optimizing Cluster Assignment

In the Section 4, we have seen that regardless of how many clusters we have, Constellation achieves a diameter of 2. Additionally, we have seen that, in the symmetric case in which all organizations have the same threshold, we can easily compute a number of clusters k such that a) validator degree is close to optimal (Theorem 1) and b) the topology is resilient to any failure admissible under the nodes' failure assumptions (Theorem 2).

We now discuss the non-symmetric case, and we present an algorithm to assign organizations to clusters so as to achieve our goal of minimizing node degree in the overlay even if the thresholds are not all the same. The resulting algorithm is the full Constellation algorithm.

Recall from Section 3 that Constellation first conservatively approximates the regular FBAS under consideration to a single-universe, regular FBAS, where each validator $v \in N$ has the universe $\mathcal{T}_v = \mathcal{O}$ (the set of all organizations) and a threshold $t_v = t_O$, where O is v's organization, with $0 < t_O \leq |\mathcal{O}|$. So we now consider such a single-universe, regular FBAS.

Remember that once each organization is assigned to a cluster, all parameters of the Constellation template are determined and the overlay is determined. Our goal is to find a partition of the organizations into clusters so that the resulting

constellation overlay has minimal average degree (Constellation guarantees a diameter of 2 in any case).

We find an optimal clustering using a simple brute-force approach to enumerate all possible partitions of the organizations and select one that a) is FBA-resilient and b) has lowest average degree. To make the problem easier, we assume that the only difference between organizations, as far as Constellation is concerned, is their threshold, meaning that any two organizations with the same threshold are interchangeable (this is true if they all run the same number of nodes). In this case, enumerating all partitions of organizations is thus the same problem as generating all partitions of a multi-set where each element is a threshold and its multiplicity is the number of organizations that have the threshold.

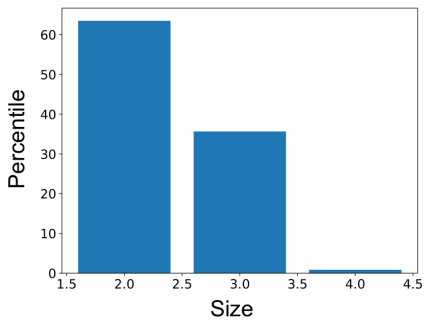
To enumerate all multi-set partitions, we implement an algorithm described by Knuth as Algorithm M [23, page 429]. However, in the worst case the number of multi-set partitions grows almost factorially[14] with the number of elements in the multi-set. To mitigate this growth, our implementation allows to limit the search space by restricting the search to a given maximum number of clusters and to a given minimum cluster size. We can also reduce the number of partitions to explore by decreasing the threshold of some organizations (which will cause them to require more connections, and thus is safe) in order to reduce the number of distinct thresholds. In a deployment of Constellation, the participating nodes would have to agree on those parameters beforehand and potential simplification of the FBAS.

G Attacks

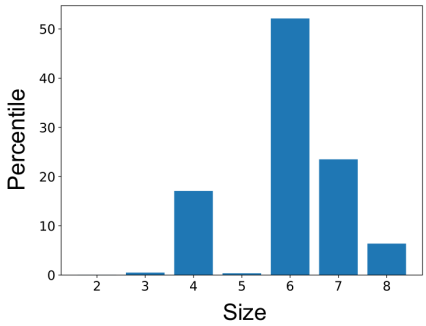
Inactive Nodes One of the simplest attack that an attacker that controls a number of nodes can perform is to simply stop forwarding traffic. To evaluate the robustness of constellation to this attack, we generate random FBA systems and simulate the attack by randomly picking a minimal quorum, removing its complement (which is a maximal allowed faulty set according to FBA) from the graph, and then determining how many more nodes, at minimum, must be removed to disconnect the graph. The results appear Figure 9. They show that, even after removing a maximal set of nodes as allowed by the FBA model, all networks maintain connectivity and exhibit robustness with large node cuts.

Maliciously-Crafted Universes and Thresholds Recall that node operators must register their FBA configuration (organization, universes, and threshold) on the blockchain and that Constellation and that the Constellation algorithm computes an overlay topology based on this information. While the consistency properties of the blockchain ensure that all nodes agree on the set of registered nodes, organizations, and their configurations, Byzantine operators are free to register configurations crafted to have maximal negative impact on the overlay

[14] An upper bound is the Bell number.



(a) Size of the node cuts after maximum crashes in 7-org network



(b) Size of the node cuts after maximum crashes in 16-org network

Fig. 9: Size of the node cuts after maximum crashes

computed by Constellation. For each organization controlled by a Byzantine operator, the operator can a) craft the universe of the organization's nodes and b) craft their threshold.

Registering large universes is particularly problematic because this can increase the size of the top-tier arbitrarily, which causes well-behaved nodes to establish a large number of connections to Byzantine nodes. Fortunately, this attack only causes additional overlay connections, and not a loss of connectivity.

A mitigating factor is that only a Byzantine organization that is part of the top-tier can increase the size of the top-tier by changing its configuration. Thus, the system is protected against such an attack as long as existing top-tier organizations remain well-behaved. Moreover, a possible defense is for the well-behaved top-tier nodes to monitor changes in registered configurations and raise an alarm e.g. if a node increases its universe by more than some percentage of the average universe size. Once an organization is suspected to act maliciously, it can be contacted off-band (in the real world) to determine whether the change in configuration is legitimate, and, if not, other node operators can remove it from their universe; if enough node operators do so, the misbehaving organization will effectively be kicked out of the top tier.

Let us now examine the impact of Byzantine nodes crafting their thresholds in order to increase the average degree of well-behaved nodes. Intuitively, to cause a maximum increase in average degree, the Byzantine nodes should set their threshold t as low as possible, meaning $|\mathcal{O} - t + 1|$ is maximized. To check this intuition, we perform simulations on a single-universe, regular FBAS consisting of 11 organizations. First, we pick 10 different minimal quorums; then for each minimal quorum, we set the threshold of the members of its complement (which is a maximal faulty sets as allowed by FBA) to the same value t, we compute the Constellation overlay, and we finally note the average degree of the members of the quorum. The results appear in Appendix G and confirm our intuition. We can see that, to negatively affect the average degree of the

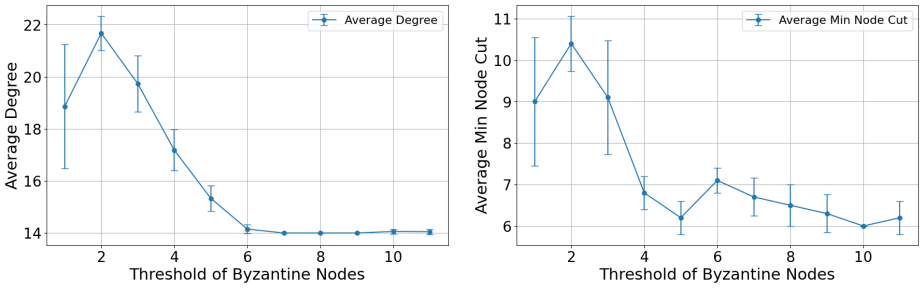


Fig. 10: Average degree and minimum cardinality of node cuts, depending on the threshold used by Byzantine nodes.

overlay, Byzantine nodes must set their threshold below 5. Luckily, in practice, well-behaved organizations are expected to set their threshold close to $2\mathcal{O}/3 \geq 7$ in order to obtain some Byzantine fault tolerance, and so thresholds below 5 should raise alarms. Appendix G depicts the average node cuts obtained in the same experiments, which show a similar pattern.

Censorship Attacks To disrupt overlay reconfiguration, Byzantine nodes can try to censor the transactions that nodes use to register or update their configuration on the blockchain. In the Stellar network, the transaction sets that are executed every 5 seconds are proposed by a leader node which changes every time and is chosen pseudo-randomly. Given that the reconfiguration time that we propose, i.e. one day, is much larger than 5 seconds, it is very likely that enough honest leaders get to propose transaction sets and ensure inclusion of the quorum set-registration transactions in the blockchain.

Resource-Exhaustion Attacks In a permissionless environment like the Stellar network, we cannot allow anyone to register a onfiguration on the blockchain for free, as this opens up the system to attacks in which malicious nodes overload the system. Charging fees or using other monetary incentives can mitigate this, but for a service as crucial as overlay reconfiguration this is unlikely to offer sufficient protection unless the funds at stake are very large. Instead, in the FBA setting, we would like to rely on the trust assumptions to mitigate this type of attack.

A possible mitigation is to require nodes to reject and not vote for transaction sets containing quorum set-registration transactions from nodes that are not in the top-tier. Of course, in this case, to compute the top-tier, nodes cannot rely on the on-chain quorum sets, as newly joining nodes are by definition not registered yet. Instead, to compute the network top-tier, nodes can use the FBA configuration information that all nodes gossip by default on the overlay in the Stellar network.