Censorship Resistance vs Throughput in Multi-Proposer BFT Protocols

Fatima Elsheimy*
fatima.elsheimy@yale.edu
Yale University
New Haven, CT, USA

Ioannis Kaklamanis*
giannis.kaklamanis@yale.edu
Yale University, IC3
New Haven, CT, USA

Sarisht Wadhwa*
sarisht.wadhwa@duke.edu
Duke University
Durham, NC, USA

Charalampos Papamanthou
charalampos.papamanthou@yale.edu
Yale University
New Haven, CT, USA

Fan Zhang
f.zhang@yale.edu
Yale University, IC3
New Haven, CT, USA

Abstract

Censorship resistance and high throughput are two key benefits of modern multi-proposer BFT protocols (such as Aptos and Sui). However, in existing designs these two properties are at odds: censorship resistance is typically achieved through duplicating transactions, which in turn harms throughput. This leaves open the question of whether it is possible to improve both properties simultaneously.

In this paper, we formally study the trade-offs between censorship resistance and throughput in multi-proposer BFT protocols, where up to f parties may be Byzantine. We present a model for the transaction assignment process, which allows us to classify assignment protocols into meaningful categories.

Using this model, we establish fundamental tradeoffs between censorship resistance and throughput. We show that under well-defined conditions, any deterministic transaction assignment protocol that achieves optimal throughput must suffer from f rounds of censorship delay; any deterministic assignment protocol that guarantees every transaction is committed within a constant number of rounds must suffer a factor of f loss in throughput relative to the optimal baseline.

On the positive side, we propose and analyze new transaction-assignment protocols that enable flexible choices among throughput–censorship tradeoffs spanning the full spectrum dictated by our lower bounds. In particular, we give a protocol that achieves $\log f$ censorship delay while paying only a factor-2 throughput loss relative to the state-of-the-art MirBFT (EuroSys'23), which incurs f rounds of censorship delay. We further propose randomized assignment protocols that provably break both the deterministic lower bound for the censorship delay and throughput in expectation. All assignment protocols can be integrated with existing multi-proposer protocols as add-ons without modifying the consensus.

1 Introduction

Blockchain networks aspire to be credibly neutral platforms where all users can freely transact without fear of censorship. In practice, however, transaction censorship has become a growing concern. Validators, builders, or relays may selectively exclude transactions because of regulations, financial incentives, or adversarial intent. Recent data indicates that **34.98% of Ethereum builders and 50% of relays engage in some form of transaction censorship** [10], demonstrating a systemic vulnerability that threatens the core principles of decentralization and neutrality.

One natural response has been to decentralize block building itself. Instead of relying on a single proposer, modern BFT-based blockchain designs increasingly use *multi-concurrent proposer (MCP)* architectures in which several proposers construct blocks in parallel; concrete examples include Mir-BFT [24], Braid [1], RedBelly [22], Solana [13] as well as DAG-based consensus deployments such as Sui and Aptos [4, 19, 20]. With multiple proposers, censorship becomes harder: no single proposer controls transaction inclusion, and ideally, a transaction needs only reach one honest proposer to be included. Consequently, an adversary that seeks to censor a transaction must bribe or corrupt multiple proposers, rather than a single leader, substantially increasing the cost and coordination required for censorship. This also weakens the serial monopoly over inclusion and ordering in traditional single-proposer blockchains, which underlies many forms of maximal extractable value (MEV).

At the same time, multi-proposer designs introduce a new tension: *transaction duplication*. Transactions must be disseminated widely to provide censorship resistance, but disseminating the same transaction broadly creates *duplication*: multiple proposers expend effort proposing overlapping transaction sets, and scarce block space is consumed by redundant inclusions rather than new transactions. This duplication directly reduces *throughput* measured in committed *unique* transactions. In today's systems, overlap is especially pronounced because proposers often select from a similar pool of pending transactions and are drawn to the same high-fee opportunities, leading to redundant inclusions.

In practice, some deployed MCP systems [5, 7, 15] rely on a shared, open mempool in which transactions are gossiped across the network and each proposer independently selects which transactions to include; others [4, 20, 22] additionally make use of RPC nodes or gateways to relay transactions between users and proposers. In the latter setting, RPC nodes often implement a system-wide "deduplication policy," deciding how many proposers should receive each transaction. While this can simplify submission for users, it does not eliminate the underlying tension: forwarding a transaction to more proposers improves its chance of reaching an honest proposer, but increases duplication; restricting forwarding reduces duplication, but makes censorship easier. Moreover, because a small number of relayers effectively control which proposers learn which transactions, they become natural censorship

*Authors contributed equally to this research, listed alphabetically.

bottlenecks: a relayer can delay or discard specific transactions even if many proposers would otherwise be willing to include them.

This inherent tension—between censorship resistance (replicating a transaction to many proposers) and high throughput (avoiding excessive duplication)—is reflected in existing protocol designs. At one extreme lie schemes such as MirBFT's hash-based bucket assignment. In MirBFT, each transaction is deterministically mapped to a unique proposer (or leader) based on a public hash function, so that in each round only one proposer is responsible for including a given transaction. This design avoids duplication entirely and therefore achieves essentially optimal throughput. However, it also offers weak censorship guarantees: under the partitioned assignment, an adversarial leader can control a transaction until the next rotation, causing delays of up to $\Theta(f)$ rounds before it is assigned to an honest proposer [24]. At the other extreme, "Full Duplication" approaches [3] (e.g., parallel chains with union-of-blocks semantics as in BRAID [1]) effectively assign each transaction to at least $f + 1$ proposers, ensuring that each transaction is proposed by an honest party in every round but suffering a multiplicative blow-up in bandwidth and throughput.

Orthogonal to multi-proposer architectures, recent work has proposed *committee-based inclusion lists* as censorship resistant add-ons that can be layered on top of existing consensus protocols. FOCIL [25] and AUCIL [26] introduce random committees whose members publish lists of mempool transactions that must be incorporated into subsequent blocks, backed by penalties or slashing for non-compliant proposers. These designs provide strong censorship guarantees, but at the cost of additional communication, latency, and protocol complexity (e.g., two rounds of reliable broadcast per slot in AUCIL). Moreover, because committee members construct their lists from local mempool views, high-value transactions (and thus likely censorship targets) may appear in multiple input lists.

Taken together, these approaches highlight the inherent tension; however, there is currently no unified framework that (i) captures the transaction assignment layer abstractly, (ii) defines censorship resistance and throughput in a comparable way across protocols, and (iii) characterizes the fundamental limitations and optimal designs in this space. In this work, we isolate the *transaction assignment* process as an explicit, modular layer sitting immediately "above" the mempool and "below" any multi-proposer BFT consensus protocol. Conceptually, an assignment protocol specifies, in each round, which subset of proposers is responsible for each mempool transaction; the underlying consensus layer is treated as a black box that guarantees that blocks proposed by honest proposers are eventually committed. This viewpoint lets us ask:

> *Can we design transaction assignment protocols that simultaneously guarantee strong censorship resistance and high throughput in a multi-proposer BFT setting? What are the best achievable trade-offs, and where do existing protocols sit relative to these limits?*

From a systems perspective, we seek *modular, consensus-agnostic gadgets* that can be plugged into existing multi-proposer BFT protocols (e.g., MirBFT, BRAID, BullShark) without changing the consensus logic. From a theory perspective, we aim to understand whether there are information-theoretic or combinatorial barriers that prevent us from achieving both "low censorship delay" and "high throughput" with deterministic assignments, and whether randomness can fundamentally improve this trade-off.

1.1 Our Contributions

We develop a general framework for studying censorship resistance and throughput in multi-proposer BFT protocols, and use it to derive tight lower bounds, improved deterministic constructions that outperform the state of the art, and randomized protocols that circumvent these bounds in expectation.

Formal Model of Transaction Assignment (Section 2). We introduce a clean formalization of *assignment protocols* as modular add-ons that, given a shared mempool, output in each round a mapping from transactions to proposers. Our model captures both deterministic and randomized assignments and distinguishes protocols based on whether the assignment logic can depend on a transaction's history—such as how long it has remained uncommitted (*transaction-reactive* vs. *transaction-non-reactive*)—or on proposer reputations (*proposer-reactive* vs. *proposer-non-reactive*). Most existing designs fall into the class of deterministic, proposer-non-reactive, transaction non-reactive protocols, including MirBFT-style schemes [24].

Metrics and Positioning of Existing Protocols (Section 3). Within this framework, we define two quantitative metrics. The *censorship resistance score* $\mathsf{CR} \in [0, 1]$ measures the worst-case delay—over all adaptive Byzantine strategies—before a targeted transaction is committed, relative to an ideal baseline of immediate inclusion (within one round of assignment to some proposer(s)). The *throughput score* $\mathsf{THR} \in [0, 1]$ measures the average number of *unique* committed transactions relative to an ideal no-duplication baseline in which all proposers are honest. More formally, the ideal baseline throughput is $n \cdot B$, where n denotes the number of proposers and B the block size of each proposer. Instantiating these metrics for existing protocols illustrates the extremes of this trade-off. For example, MirBFT assignment achieves near-optimal throughput score, with $\mathsf{THR} = \frac{n-f}{n}$, as every honest proposal is unique and up to f blocks are Byzantine. However, it guarantees weak censorship resistance score, with $\mathsf{CR} = O(1/f)$, as a transaction may wait $O(f)$ rounds before being assigned to an honest proposer. At the opposite extreme, a Full-Duplication scheme that assigns each transaction to $f + 1$ proposers in every round guarantees a constant censorship resistance score, with $\mathsf{CR} = O(1)$, but incurs throughput score $\mathsf{THR} = O(1/f)$ due to factor of $O(f)$ duplication. These protocols lie at opposite extremes of the censorship–throughput trade-off.

Deterministic Lower Bounds (Section 4). Our main negative result proves the impossibility of avoiding this trade-off with deterministic assignment protocols. Under standard BFT scaling ($n = \Theta(f)$), we formally prove that any deterministic protocol achieving a constant censorship resistance score ($\mathsf{CR} = O(1)$, i.e., any transaction is committed within a constant number of rounds after being assigned to some proposer) must incur throughput score $\mathsf{THR} = O(1/f)$, i.e., a factor of f less than ideal throughput. Conversely, any deterministic protocol that achieves near-optimal throughput $\mathsf{THR} = 1 - O(f/n)$ must tolerate censorship delays of $O(f)$ rounds, corresponding to $\mathsf{CR} = O(1/f)$. As a consequence,

no deterministic assignment protocol can simultaneously achieve constant censorship delay and optimal throughput.

The lower bounds rely on two complementary adversarial arguments. For protocols with constant censorship delay, we show that any transaction must be assigned to at least $f+1$ distinct proposers within a constant window of rounds; otherwise, an adversary can corrupt exactly those proposers and delay commitment beyond the constant delay. This necessarily induces $O(f)$ duplication in some round, implying a $\Theta(1/f)$ throughput loss (Theorems 4 and 5). Conversely, for protocols achieving near-optimal throughput, we exploit the fact that minimal duplication must occur over many rounds; an adversary can then target a transaction by corrupting its assigned proposer in each round, yielding an $\Omega(f)$ censorship delay (Theorems 6 and 7). Figure 1 shows the infeasible region dictated by our lower bounds, shown in gray. For $n = 3f + 1$, worst-case throughput is fundamentally limited to $(n - f)/n \approx 2/3$; we show in Theorem 1 that this bound is tight under a worst-case strategy.

Transaction-Reactive Protocols (Section 5). On the positive side, we design deterministic assignment protocols that achieve improved trade-offs within the limits imposed by our lower bounds. We begin with a simple *k-duplication* scheme (blue curve in Figure 1), in which each transaction is assigned to k proposers per round. Increasing k improves the censorship resistance score but proportionally reduces throughput due to duplication. We then introduce *transaction-reactive* deterministic protocols that adjust a transaction's duplication factor over time. Rather than using the same duplication factor for all transactions, these protocols increase a transaction's duplication factor if it remains uncommitted, concentrating duplication on transactions that appear to be censored. We study two concrete families based on additive-increase and multiplicative-increase, and show that they realize a broad spectrum of achievable (CR, THR) trade-offs between the deterministic lower bounds. In particular, a carefully tuned multiplicative-increase protocol (red curve in Figure 1, with initial duplication factor $k = 1$ and duplication growth factor 2) achieves an $O(\log f)$ censorship delay while preserving throughput within a factor 2 of MirBFT. By contrast, MirBFT achieves throughput score 2/3 but suffers $\Theta(f)$ censorship delay (see Table 1), so our protocol strictly improves delay at comparable throughput. The MI curve is capped at throughput score 1/3, since any transaction censored in round r is duplicated to twice as many proposers in round $r+1$, yielding an average 2× duplication overhead (closed-form scores in Section 3.1 and Section 5.1).

Randomized Assignments Beyond Deterministic Limits (Section 6). Finally, we study randomized assignment protocols in which the transaction-to-proposer mapping in each round is sampled from a distribution that is unpredictable to the adversary (e.g., via a common coin). These protocols follow the k-duplication structure—each transaction is assigned to k proposers per round—but replace the fixed assignment rule with randomized sampling, preventing the adversary from conditioning its strategy on the realized assignment of that round. We analyze two concrete instantiations. The first assigns each transaction to exactly k random proposers per round, yielding exact duplication k but only an expected per-proposer block size B. For the second protocol, each proposer independently samples exactly B transactions from a shared

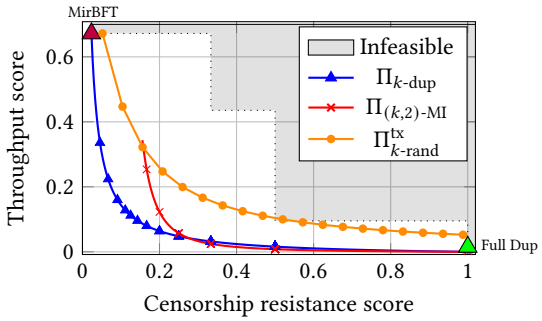

Figure 1: Throughput–censorship tradeoff. The gray region shows infeasible throughput–censorship tradeoffs excluded by our lower bounds. Its boundary appears step-wise because the censorship resistance score takes the form $1/D$ for integer D. The blue, red, and orange curves correspond to the deterministic, improved deterministic, and randomized assignment protocols, respectively. Consistent with the deterministic setting, the blue and red curves pass only through points with $\mathrm{CR} = 1/D$ for integer D; the randomized protocol is not constrained since D is a real-valued function based on the tail bound considered, so the orange curve can attain intermediate values. Parameters are $n = 3f + 1$ with $f = 42$.

candidate pool of size nB/k. The size of this pool reflects system capacity: across n proposers, each proposing B transactions with expected duplication k, at most nB/k distinct transactions can be assigned per round. For both designs, we prove that for constant k the protocols achieve constant throughput score ($\sim 1/k$) with high-probability concentration around the expectation (via Chernoff bounds), together with $O(\log(f))$ censorship delay with high probability. Similarly, randomized assignment achieves optimal censorship resistance score 1 with throughput score $O(1/\log f)$, losing only a logarithmic factor relative to ideal (orange curve in Figure 1). Consequently, randomized schemes match the throughput of deterministic protocols while significantly improving censorship delay, thereby circumventing our deterministic lower bounds.

Table 1 provides key points in the comparison between our constructions and selected existing constructions in the literature. In Figure 1, we plot all protocols with $n = 128$ proposers, $f = \lfloor \frac{1}{3} \cdot 128 \rfloor = 42$ Byzantine proposers, and per-proposer block size $B = 5$. These parameters match real-world deployments such as Sui [20] and Aptos [4], and we use them throughout the paper.

2 Formal Foundations

We formalize our system model in the classical BFT setting and introduce a framework for *assignment protocols*. Within this framework, we classify assignment protocols into meaningful categories and analyze fundamental lower bounds on their performance.

[1] DAG-based protocols. Censorship delay depends on client dissemination: wide dissemination (e.g., $> 2f$ proposers) yields constant delay, while limited dissemination (e.g., $O(1)$ proposers) can incur $O(f)$ delay.
[2] It is reported that Sui [20] duplicates each transaction up to five times to increase inclusion probability.
[3] For AUCIL, the bound assumes the adversary lacks sufficient bribery budget to sustain censorship across committee rotations.

Protocol	TX Censorship Delay	Throughput Loss Factor
Single-leader BFT [8, 27], MirBFT [24]	f	1
DAG-based protocols [4, 6, 20, 23]	$1 \to f$ [1]	$f \to 1$
e.g. Sui [19]	$f/5$ [2]	5
FOCIL [25]	1	f
AUCIL [26]	Based on fee paid (1^3)	Based on fee distribution
$k-$Duplication (Section 3.1)	$1 \to f$	$f \to 1$
$(k, \ell)-$Multiplicative Increase (Section 5.1)	$1 \to O(\log f)$	$f \to 2$
$k-$Randomized (Section 6)	$1 \to O(\log f)$	$O(\log f) \to 1$

Table 1: Comparison of our work to existing literature on censorship and duplication-resistant BFT protocols. *TX Censorship Delay* measures the worst-case number of rounds an adversary can delay the commitment of any transaction. *Throughput Loss Factor* measures the multiplicative throughput reduction compared to fully unique honest proposals. If protocol Π has delay $a \to b$ and loss factor $c \to d$, this means that Π achieves delay a with loss c and delay b with loss d. Moreover, a and b are the extreme delays achieved by Π.

2.1 Definitions and Main Components

We focus on assignment protocols that can be seamlessly "plugged in" immediately before a multi-proposer BFT consensus protocol. The role of the assignment protocol is to distribute transactions from the mempool among proposers, who then input their assigned transactions into consensus. The consensus protocol itself is treated as a black box that takes as input the set of transactions assigned per-proposer and outputs a set of committed transactions.

While the underlying consensus protocol may span multiple communication steps (e.g., message rounds), we abstract these details away: in each round r of our model, an entire *view* of the consensus mechanism is assumed to execute and produce a committed set.

We focus on assignment protocols that are *non-reactive with respect to proposers* – that is, the protocol treats all proposers identically, regardless of their past behavior. In particular, if a proposer misbehaves or fails to propose the assigned transactions, the assignment mechanism does not penalize or exclude them in future rounds. Although incorporating reactivity or slashing is an interesting direction for future work, such mechanisms are more naturally studied in the *rational* setting. In contrast, our analysis focuses on the classical *Byzantine* model, where the non-reactive class already captures natural and expressive assignment protocols that provide meaningful insights for blockchain designers when selecting assignment strategies.

Setup and Network Assumptions. We consider a set of proposers denoted by $\mathcal{P} = \{P_1, \ldots, P_n\}$. We assume the standard Byzantine setting where the number of proposers satisfies $n = cf + d$ for some constants $c, d > 0$, and f denotes the adversary's corruption budget. We leave c and d unspecified to encompass different BFT configurations, such as $n = 3f + 1$ [27] and $n = 5f + 1$ [18].

As discussed earlier, we consider a multi-proposer BFT setting in which multiple consensus instances run concurrently. Specifically, the underlying consensus layer may be realized either as n parallel chains of single-leader BFT protocols [16, 27] or as an off-the-shelf multi-concurrent BFT protocol [23]. We treat this consensus layer as a black box that guarantees the standard BFT properties of *safety* and *liveness*. In particular, we assume that any block proposed by an honest proposer is eventually committed—an assumption that holds for any protocol that is tail-forking resistance, such as [14, 16]. Each proposer can propose a block of size at most $B \in \mathbb{N}$.

Finally, for all assignment protocols studied in this work, we assume a pre-arranged coordination mechanism that allows proposers to deterministically compute the assignment and instantly learn which transactions they are responsible for proposing, without incurring additional latency or requiring any interaction among parties. To enable this, we assume that all proposers maintain a consistent shared view of the mempool and execute the assignment algorithm locally over this shared view.

Adversarial Model. We consider a static, unbounded adversary $\mathcal{A}$ that can corrupt $\mathcal{F} \subset \mathcal{P}$ proposers, $|\mathcal{F}| \leq f$, who may deviate arbitrarily from the protocol. When assigned a set of transactions to propose, the adversary may choose to propose all of the transactions, a subset, or none at all. For deterministic assignment protocols, the adversary can simulate the protocol's execution in advance and therefore fix its strategy $\mathcal{A}_r$ for every round r ahead of time (i.e., before round 1). For randomized assignment protocols, it is crucial that the randomness used in each round r is both uniform and unpredictable prior to its revelation. In particular, no adversary can predict the random string of round r before it is revealed. Consequently, the adversary must commit to its strategy $\mathcal{A}_r$ before observing the randomness for that round. We use $\mathcal{A}$ to denote the collection of all per-round strategies adopted throughout the execution, and $\mathcal{A}^*$ to denote the set of all possible adversarial strategies.

2.2 Assignment Protocol

We provide a formal definition of an assignment protocol as follows:

Definition 1 (Assignment Protocol)**.** *An* assignment protocol Π *is a protocol that, in each round r, produces an assignment*

$$A_r \leftarrow \Pi(\mathcal{P}, r, M_r, \mathsf{Hist}_r),$$

where $\mathcal{P}$ is the set of proposers, $M_r = [\mathsf{tx}_1, \ldots, \mathsf{tx}_k]$ denotes the ordered mempool at round r (where each $\mathsf{tx}_\ell \in \mathcal{T}$ is drawn from the global transaction universe $\mathcal{T}$, and $k \in \mathbb{N}$ denotes the mempool size), and Hist_r is an external state variable that may encode auxiliary metadata about previous rounds (with $\mathsf{Hist}_0 = \emptyset$). The output is a mapping $A_r : M_r \to 2^{\mathcal{P}}$ that assigns transactions in the mempool to a subset of proposers responsible for proposing it in round r.

Duplication Factor. We define duplication factor as the number of proposers a transaction is assigned to in a given round. For any protocol Π, the number of times a transaction tx repeats in the assignment A_r is denoted by $d(\mathsf{tx}, \mathcal{P}, r, M_r, \mathsf{Hist}_r) = \mathrm{count}(\mathsf{tx}, A_r = \Pi(\mathcal{P}, r, M_r, \mathsf{Hist}_r))$. When using $d()$, we abuse notation by including only the arguments that are relevant to the context in question.

2.2.1 Classifying Assignments. Using our framework, we classify assignment protocols depending on whether they *react* based on proposers' behavior or based on transactions not being committed.

Proposer Reactivity. Informally, an assignment protocol is *proposer*-non-reactive if it treats all proposers the same, regardless of their past behavior. To define this property formally, we will consider the output of the assignment protocol under arbitrary permutations of the proposer list.

Formally, for any permutation (bijection) $\sigma : [n] \to [n]$, let $\sigma(\mathcal{P}) := [P_{\sigma(i)}]_{i=1}^{n}$ be the permuted proposer list of $\mathcal{P} = [P_i]_{i=1}^{n}$ with respect to $\sigma(\cdot)$.

Definition 2. *An assignment protocol* Π *is* proposer-non-reactive *if and only if* $\forall \sigma : [n] \to [n]$

$$\forall r, M, \mathsf{Hist} : \Pi(\mathcal{P}, r, M, \mathsf{Hist}) = \Pi(\sigma(\mathcal{P}), r, M, \mathsf{Hist}).$$

Otherwise, the protocol is proposer-reactive.

Since all proposers are treated symmetrically in every round, any permutation of proposer identities leaves the assignment unchanged. Hence, for a *proposer-non-reactive* protocol, the assignment in round r depends only on the current state and not on prior proposer actions; the adversary's past behavior is irrelevant.

Transaction Reactivity. Informally, an assignment protocol is *transaction*-non-reactive if it treats all transactions the same, regardless of their history. Equivalently, such a protocol does *not* make use of the external state Hist.

Definition 3. *An assignment protocol* Π *is* transaction-non-reactive *if and only if*

$$\forall r, M, \mathsf{Hist}, \mathsf{Hist}' : \Pi(\mathcal{P}, r, M, \mathsf{Hist}) = \Pi(\mathcal{P}, r, M, \mathsf{Hist}').$$

Otherwise, the protocol is transaction-reactive.

As a consequence, the duplication factor of any transaction in assignment is independent of Hist in previous rounds for any *transaction-non-reactive* protocol.

2.3 Execution Components

In this section, we describe the system components that interact with the assignment protocol.

2.3.1 Environment. The environment $\mathsf{Env}_{\mathcal{A}}$ models the execution of the underlying multi-proposer consensus protocol as well as the refilling of new transactions into the mempool. Intuitively, $\mathsf{Env}_{\mathcal{A}}$ serves as the interface between the assignment protocol and the consensus layer: it takes as input the transaction assignment A_r for round r and determines, based on the underlying consensus protocol and the adversarial strategy, which transactions are successfully committed by the end of the round. At the same time, it generates a fresh set of transactions to refill the mempool for the next round.

Formally, given an assignment A_r and adversarial strategy $\mathcal{A}$, the environment produces

$$(C_r, N_{r+1}) \leftarrow \mathsf{Env}_{\mathcal{A}}(A_r),$$

where:

- $C_r \subseteq A_r$ denotes the subset of transactions that were assigned and *committed* by the end of round r; and
- $N_{r+1} \subseteq \mathcal{T}$ denotes the set of new transactions introduced by clients to refill the mempool for round $r + 1$.

Conceptually, $\mathsf{Env}_{\mathcal{A}}$ captures all environmental factors that are external to the assignment protocol itself: the consensus protocol, the adversary's strategy, network delays, and the arrival of new transactions. This abstraction allows us to reason about the censorship and throughput properties of an assignment protocol independently of the consensus mechanism.

Assumption: Honest Inclusion. We impose the following assumption on the behavior of the environment $\mathsf{Env}(\cdot)$. For every round r, if the assignment A_r maps a transaction $\mathsf{tx} \in M_r$ to at least one honest proposer $P \in \mathcal{H}$, then tx is guaranteed to be committed in that round:

$$\forall\, r \geq 1,\ \forall\, \mathsf{tx} \in M_r,\ \Big(\exists\, p \in \mathcal{H} : p \in A_r[\mathsf{tx}]\Big) \implies \mathsf{tx} \in C_r.$$

This captures the desirable property that censorship only arises from assignments that exclusively map transactions to corrupted proposers. As a direct consequence of this assumption, the number of assigned transactions that fail to be committed in round r is upper-bounded by the number of corrupted proposers, i.e., $|A_r \setminus C_r| \leq f$.

2.3.2 Mempool Structure. We model the mempool M as a set that maintains all uncommitted transactions from which the assignment protocol selects transactions. Once a transaction is committed, it is removed from the mempool, which is then refilled with newly arriving transactions.

Conceptually, M should retain *some* information about which transactions remain uncommitted—so that they are prioritized in subsequent rounds, as further discussed in Section 2.4—while revealing *as little* as possible about the assignment process itself. In particular, the mempool should not reveal whether a transaction has already been assigned but not yet committed. It should remain agnostic to the internal state or history of the assignment protocol and maintain no information about previous assignments.

We define the algorithm UpdateMempool, which updates the mempool and the external state, Hist after each round. Given the current mempool M, the set of newly committed transactions C, and the set of newly arrived transactions N, the algorithm removes C from M, adds N, and randomly permutes the remaining transactions while keeping them at the top of the mempool list. It also updates the external map Hist′: for each surviving transaction, it represents the number of rounds in which the transaction was assigned but not yet committed. Each new transaction in N is initialized with counter 0. The algorithm outputs the updated pair (M', Hist').

Relation to Assignment Classification. Per Definition 3, transaction-reactive protocols may depend on the external state Hist, whereas transaction-non-reactive protocols remain independent of it. As seen in line 4, the UpdateMempool procedure ensures that transactions assigned but not yet committed are prioritized at the top of the

Algorithm 1 UpdateMempool

```
function Update(M, A, C, N, Hist)
    L' ← { tx ∈ A | tx ∉ C }
    Randomly permute L'
    M' ← L' | (M \ A) | N
    for all tx ∈ L' do
        if tx ∈ A then
            Hist[tx] ← Hist[tx] + 1
    for all tx ∈ N do
        Hist[tx] ← 0
    return (M', Hist)
```

mempool in subsequent rounds. This design allows transaction-non-reactive protocols—despite their lack of access to Hist—to continue including uncommitted transactions L in future assignments without explicitly distinguishing them from newly arrived transactions.

Here, "include" does not mean assigning higher priority or preference to leftover transactions, as that would imply reactivity. Rather, it means that when the protocol selects a total of $W < nB$ (or $W < |L|$) transactions, it is permitted to fill this quota with transactions from L if it so happens, without inadvertently excluding some of them in favor of newer transactions. At the same time, we wish to avoid revealing any information that might enable even a non-reactive protocol to prioritize certain transactions in L over others, or to distinguish between L and $M \setminus A$. To achieve this, the algorithm applies a random permutation to L and, crucially, does not expose the boundary between leftover and unassigned transactions within the mempool.

Execution Trace. An execution trace captures the evolution of the system state. Each round proceeds in three steps: (1) the assignment protocol selects transactions to assign to proposers, (2) the environment determines which transactions are committed and which new ones are added to the system, and (3) the mempool and external state are updated accordingly.

Definition 4 (Execution Trace). *Let Π be an assignment protocol. Fix a number $s \in \mathbb{N}$ and an adversary strategy $\mathcal{A}$. We define the* execution trace *of Π for s rounds under $\mathcal{A}$ to be $\mathsf{Tr}^{\mathcal{A}}_{\Pi}(s) := \{M_r^{\mathcal{A}}, A_r^{\mathcal{A}}, C_r^{\mathcal{A}}\}_{r\in[s]}$, where for each $r \in [s]$,*

$$(A_r) \leftarrow \Pi(\mathcal{P}, r, M_r, \mathsf{Hist}_r),$$
$$(C_r, N_{r+1}) \leftarrow \mathsf{Env}_{\mathcal{A}}(A_r),$$
$$(M_{r+1}, \mathsf{Hist}_{r+1}) \leftarrow \textsc{Update}(M_r, A_r, C_r, N_{r+1}, \mathsf{Hist}_r).$$

Here $M_1 := \mathsf{Env.Init}()$ and $\mathsf{Hist}_1[\mathsf{tx}] = 0$ for all $\mathsf{tx} \in M_1$.

Initialization. For the first round ($r = 1$), we set:

- $N_1 \leftarrow \mathsf{Env.Init}()$, where $\mathsf{Init}()$ is an assumed initialization function of the Environment; outputs the first set of transactions.
- $(M_1, \mathsf{Hist}_1) \leftarrow \textsc{Update}(\emptyset, \emptyset, \emptyset, N_1, \emptyset)$.

2.4 Valid Assignment Protocols

We say any assignment protocol Π is valid if, for all $\mathcal{A} \in \mathcal{A}^*$, the execution trace $\mathsf{Tr}^{\mathcal{A}}_{\Pi}()$ satisfies the following conditions:

(1) Respects the block size limit. For all rounds r,

$$\forall p \in \mathcal{P}: \quad \left|\{\mathsf{tx} \in M_r : p \in A_r[\mathsf{tx}]\}\right| \le B.$$

(2) An uncommitted transaction must not be re-assigned to any proposer that has already been assigned that transaction in prior rounds. Formally, for all rounds r, proposers P_j, and transactions tx_i,

$$\forall r_k \in [0, r): \quad P_j \in A_r[\mathsf{tx}_i] \implies P_j \notin A_{r_k}[\mathsf{tx}_i].$$

The rationale behind this constraint is that repeatedly assigning a transaction to the same proposer can trivially break censorship resistance; if an adversary corrupts that proposer, the corresponding transaction could be indefinitely censored. To avoid such cases, we require that once a proposer fails to include an assigned transaction, that transaction must be reassigned to a different set of proposers in subsequent rounds.

(3) Must assign the leftover transactions. Formally, If $\mathsf{tx} \in A_r$, then $\mathsf{tx} \in A_{r+1} \vee \mathsf{tx} \in C_r$. This ensures that our measure of censorship resistance reflects only adversarial behavior rather than transaction congestion. Once a transaction is assigned, it remains assigned in all subsequent rounds until committed; any delay in its commitment is solely due to adversarial interference.

(4) Balanced assignment of transactions: The duplication factor of any transaction is independent of other transactions that have been committed or assigned. Also, all previously unassigned transactions that are assigned in any round have the same duplication factor. Formally, $d(\mathsf{tx}, \_, M_r) = d(\mathsf{tx}, \_, M'_r)$ and $\forall \mathsf{tx}_1, \mathsf{tx}_2 \in A_r : \mathsf{tx}_1, \mathsf{tx}_2 \notin A_{r-1} \implies d(\mathsf{tx}_1, \_) = d(\mathsf{tx}_2, \_)$.

As a consequence of constraints (3) and (4), a protocol that attempts to increase duplication factor of all the leftover transactions to a number that cannot be accommodated (i.e., exceeds the block size) will be considered invalid. Consider an execution such that until round $r-1$, the adversary does not censor any transaction, i.e., $C_{s-1} = A_{s-1}, \forall s \le r-1$. In such a case, due to (4), the duplication factor of all transactions will be the same in round r as defined by the assignment protocol. Let this duplication factor be d. The number of transactions the adversary can censor this round can be at most $\frac{fB}{d}$. In the next round, let the duplication factor of leftover transactions be d'. Since all of these leftover transactions must be assigned in round r, $d' \frac{f}{d} \le n$. The ratio of consecutive round duplication factor is therefore $\frac{d'}{d} \le \frac{n}{f} = 1/\eta$, where η is defined as the ratio of total number of faults to the total number of proposers.

All lemmas, lower bounds, and constructions are stated with respect to the space of valid assignment protocols. Consequently, we omit explicitly specifying validity assumption in the statements.

2.5 Censorship Resistance and Throughput

We now formally define the *censorship resistance* and *throughput* of a deterministic assignment protocol within our framework.

Censorship Resistance. For a given assignment protocol, let D denote the maximum number of rounds that an adversary can delay the inclusion of any transaction in a BFT execution. We define the *censorship resistance score* as

$$\mathsf{CR} := \frac{1}{D+1} \in (0, 1].$$

Intuitively, the numerator 1 represents the ideal case where all proposers are honest—i.e., the best-case scenario in which every

transaction is committed in the same round it is assigned, corresponding to a one-round delay. Thus, CR quantifies how much the adversary can disrupt this ideal behavior: lower values indicates stronger adversarial influence, i.e., weaker censorship resistance.

We first define the realized delay $D^{\mathcal{A}}$ under a fixed adversarial strategy $\mathcal{A}$ in Definition 5, and subsequently define the maximum delay D and the censorship resistance score CR in Definition 6.

Definition 5 (Delay Under Adversary)**.** *Let* Π *be an assignment protocol and* $\mathcal{A}$ *be a fixed adversary. Let* $\{M_r, A_r, C_r\}_{r=1,\dots,\infty}$ *be the execution trace of* Π*. Let* $\mathcal{T}'$ *be the set of transactions that have been assigned during* s *rounds of the above execution;* $\mathcal{T}' = \bigcup_{r=1}^{s} A_r$*. For any transaction* $\mathsf{tx} \in \mathcal{T}'$*, we define* a_{tx} *and* c_{tx} *to be the first round that* tx *was assigned and committed respectively (*$c_{\mathsf{tx}} = \infty$*, if* tx *is never committed). Then define the* censorship delay *of protocol* Π under $\mathcal{A}$ *to be*

$$D_{\Pi}^{\mathcal{A}} = \lim_{s\to\infty}\left(\max_{\mathsf{tx}\in\mathcal{T}'}(c_{\mathsf{tx}} - a_{\mathsf{tx}})\right).$$

Definition 6 (Censorship Resistance Score)**.** *Let* Π *be an assignment protocol. We define the* censorship delay *of protocol* Π *to be the maximum delay under all possible adversarial strategies,* $\mathcal{A}^*$ *i.e.,*

$$D_{\Pi} := \max_{\mathcal{A}\in\mathcal{A}^*} D_{\Pi}^{\mathcal{A}},$$

where $D_{\Pi}^{\mathcal{A}}$ *is the censorship delay of* Π *under* $\mathcal{A}$*, per Definition 5. Finally, we define the* censorship resistance score *of* Π *to be*

$$\mathsf{CR}_{\Pi} := \frac{1}{D_{\Pi}+1}.$$

Throughput. For a given assignment protocol, let T be the *average* number of committed transactions in the BFT setting, where the average is taken over rounds. Then our throughput score is $\mathsf{THR} := \frac{T}{nB} \in (0,1]$. We place the value nB in the denominator to capture the fact that, under all proposers being honest, the best achievable throughput (of any protocol) is exactly nB transactions per round. Therefore, we can interpret THR as a measure of the disruption caused by the adversary compared to the nB "ideal" throughput. We first define the realized throughput $T^{\mathcal{A}}$ under a fixed adversary $\mathcal{A}$ in Definition 7, and subsequently define the throughput score THR in Definition 8.

Definition 7 (Throughput Under Adversary)**.** *Let* Π *be an assignment protocol, and* $\mathcal{A}$ *be a fixed adversary strategy. For any* $s \in \mathbb{N}$*, let* $\mathsf{Tr}_{\Pi}^{\mathcal{A}}(s) := \{M_r, A_r, C_r\}_{r\in[s]}$ *be the execution trace of* Π *for* s *rounds under* $\mathcal{A}$*, per Definition 4. We define the* throughput *of protocol* Π under $\mathcal{A}$ *to be*

$$T_{\Pi}^{\mathcal{A}} := \lim_{s\to\infty}\left(\frac{1}{s}\sum_{r=1}^{s}|C_r|\right).$$

Definition 8 (Throughput Score)**.** *Let* Π *be an assignment protocol. We define the* throughput *of protocol* Π *to be the minimum throughput under all possible adversarial strategies* $\mathcal{A}^*$*, i.e.,*

$$T_{\Pi} := \min_{\mathcal{A}\in\mathcal{A}^*} T_{\Pi}^{\mathcal{A}},$$

where $T_{\Pi}^{\mathcal{A}}$ *is the throughput of* Π *under* $\mathcal{A}$*, per Definition 7. Finally, we define the* throughput score *of* Π *to be*

$$\mathsf{THR}_{\Pi} := \frac{T_{\Pi}}{nB}.$$

We show that for any proposer-non-reactive assignment protocol, the adversary's throughput-minimizing strategy is to remain silent. More formalize and prove this statement in Theorem 1.

Theorem 1 (Worst-case adversarial strategy)**.** *Given a proposer-non-reactive assignment protocol* Π*. Let the adversary be static with at most* f *corruptions. Among all possible adversarial strategies, the strategy that minimizes throughput is* silence*: after selecting the* f *proposers to corrupt, denoted by set* $\mathcal{F}$*, the adversarial proposers submit no transactions in any round.*

To formally prove this, we introduce and prove the following lemma.

Lemma 1 (Per-transaction, per-round dominance of silence)**.** *Let* $\mathsf{tx} \in A_r$ *be a transaction that a corrupted proposer* $q \in \mathcal{F}$ *may declare in round* r*. Consider two adversarial strategies that are identical except that* q *declares* tx *in round* r *(*Incl*) or remains silent (*Sil*). Then for all* $r' \geq r$*, the number of transactions committed by round* r' *under* Sil *is at most that under* Incl*.*

Proof. Given M_r, A_r and honest messages in round r.
Case A: tx *is assigned and included by some other proposer in* A_r. $\mathsf{tx} \in C_r$ regardless of q's choice. Declaring tx by q cannot *reduce* commitments. Hence $|C_r|$ under Sil is at most that under Incl.
Case B: tx *is not assigned or assigned but not included by any other proposer in* A_r. If q declares tx (Incl), then the set of committed transactions is one more than in Sil.

For rounds $> r$, note that UpdateMempool (Alg. 1) updates M_{r+1} by removing C_r and randomly permuting leftovers, then appending N_{r+1}. Since C_r under Sil is a subset of that under Incl, the leftover set under Sil is a *superset* of the leftover set under Incl. Because Π is balanced (Validity condition 4), the duplication of each of the leftover transactions remains the same. The duplication of each new transaction also remains the same. The number of newly added transactions in Incl is thus at least the same as the number of newly added transactions in Sil. This implies in the next round the number of committed transactions (if all other strategy remains the same), is at most one more in Sil than Incl. Since in round r, the set of committed transactions in Incl was one more than Sil, the sum implies that the union of committed transactions in Incl is at least the same number in Sil. Inductively, the cumulative commitments up to any $r' \geq r$ under Sil are at most those under Incl. □

Proof of Theorem 1. Let σ be any adversarial strategy profile for corrupted proposers over rounds $1, 2, \dots$. We transform σ into the *silent* strategy σ^{sil}. This can be done by looking at each each corrupted proposers included transaction in each round. By replacing the included transaction with silence for that transaction, from Lemma 1, we can create a strategy with the same or worse throughput. This step is repeated until no corrupted proposer has any transaction included in its included set and thus staying silent is the best strategy for the corrupted proposer. □

Remark 9 (Best possible throughput)**.** *Given any assignment protocol* Π*, an adversary* $\mathcal{A}$ *can corrupt any* f *parties and keep them silent (i.e., have them not propose any transaction). Under* $\mathcal{A}$*, protocol* Π *can have at most* $(n-f)\cdot B$ *distinct transactions committed per round. Thus, we observe that* no *protocol can achieve a throughput score strictly greater than* $\frac{n-f}{n}$*, per Definition 8.*

3 Case Studies of Assignment Protocols

We present case studies from the literature on transaction assignment protocols and analyze their throughput and censorship resistance according to our formal definitions. We introduce a general family of assignment protocols, which we refer to as the transaction-non-reactive k-Duplication assignment family, or $\Pi_{k\text{-dup}}$ for short. This framework subsumes both the MirBFT protocol and the Full Duplication scheme discussed in Section 1 – the only two known BFT protocols that explicitly define how transactions are distributed among proposers.

3.1 Transaction Non-Reactive k-Duplication

In $\Pi_{k\text{-dup}}$, the term k-duplication indicates that each transaction is assigned to k proposers in a given round. This is a transaction-non-reactive protocol, meaning that new and uncommitted transactions are treated identically. Consequently, the duplication factor for any transaction remains constant across rounds. Algorithm 2 formally specifies $\Pi_{k\text{-dup}}$. In each round r, every transaction in the mempool is assigned to k consecutive proposers in a cyclic manner. The starting point rotates deterministically by a stride of k between rounds, ensuring that a persistent transaction gets assigned to a new proposer in every round.

Algorithm 2 k-Duplication Assignment

function $\Pi_{k\text{-DUP}}(\mathcal{P}, r, M, \text{Hist})$
  Let $n \leftarrow |\mathcal{P}|$
  $A_r := \emptyset$
  ▷ *Compute the effective number of transaction slots per round*
  $K \leftarrow \frac{n \cdot B}{k}$
  $\text{offset} \leftarrow (r \cdot k) \bmod K$
  for $i \in [K]$ **do**
    Let tx_i denote the i-th transaction in M
    ▷ *Compute the starting proposer index for this transaction's assignment*
    $\text{start} \leftarrow (\text{offset} + \text{hash}(\text{tx}_i) \cdot k) \bmod n$
    ▷ *Determine the k proposers responsible for tx_i in this round*
    $S \leftarrow \{\, (\text{start} + d) \bmod n \mid d \in [0 : k-1] \,\}$
    $A_r[\text{tx}_i] \leftarrow \{\, P_j \mid j \in S \,\}$
  return A_r

In Theorems 2 and 3, we derive upper bounds on throughput and censorship resistance per our formal definitions in Section 2.5.

Theorem 2. *The throughput score of $\Pi_{k\text{-dup}}$ is* $\text{THR} = \frac{(n-f)}{nk}$.

Proof. Let $|H| = n - f$ represent the number of honest proposers. In the k-duplication assignment protocol, each transaction is assigned to exactly k distinct proposers. Hence, across all honest proposers, the total number of transaction slots assigned per round is

$$S_H = (n-f)|B|.$$

Since each distinct transaction occupies at most k slots (one per proposer it is assigned to), the number of *distinct* transactions among the honest proposers' assignments, denoted U, satisfies

$$\text{THR}_H \le k \cdot |U|.$$

Rearranging, we obtain a lower bound on the number of distinct transactions held by honest proposers:

$$|U| \ge \frac{S_H}{k} = \frac{(n-f)|B|}{k}.$$

Next, consider the adversarial strategy. The adversary achieves minimum throughput by having the f malicious proposers withhold proposals or propose as per Theorem 1. Such a strategy eliminates their contribution but cannot reduce the number of distinct transactions assigned to honest proposers beyond the combinatorial bound above. Thus, in the worst case, only honest proposers contribute to the protocol's throughput. Consequently, the cumulative throughput over round w is

$$T^{\mathcal{F}}(w) := \frac{1}{w} \sum_{r=1}^{w} \Big| \frac{(n-f)|B|}{k} \Big|,$$

and the throughput score is

$$\text{THR} := \frac{\lim_{w\to\infty} T(w)}{nB} = \frac{(n-f)}{nk}.$$

□

Theorem 3. *The censorship resistance score of $\Pi_{k\text{-dup}}$ satisfies* $\text{CR} \le \frac{k}{f+k}$ *for $k \le f$, and* $\text{CR} = 1$ *for $k > f$.*

Proof. Fix a target transaction τ and fix an arbitrary initial round index (w.l.o.g. round 1). Because assignments rotate by a stride of k proposers each round, the set of proposers assigned to τ in round t is

$$S_t \;=\; \{\, s + (t-1)\cdot k + d \pmod n \mid d = 0, \ldots, k-1 \,\},$$

for some base index s determined by τ. By construction, each S_t contains exactly k distinct proposer indices. Moreover, for distinct rounds $t \neq t'$ the sets S_t and $S_{t'}$ are disjoint until the assignment wraps around modulo n; in particular, for the purpose of short blocking intervals (i.e., fewer than n/k rounds) the sets S_t are disjoint.

An adversary that wishes to censors τ from being proposed by any honest proposer during rounds $0, 1, \ldots, D$ must ensure that for every $t \in \{0, \ldots, D\}$ all proposers in S_t are Byzantine (otherwise an honest proposer in some round would propose τ). Because the S_t sets are disjoint for the range of rounds of interest, the adversary needs to control at least k distinct proposers per blocked round. Hence, to block D consecutive rounds the adversary must control at least kD distinct proposers. Since the adversary controls at most f proposers, we obtain the necessary inequality

$$kD \;\le\; f.$$

Therefore, the largest delay T that the adversary can guarantee until the transcation is committed satisfies

$$D \;\le\; \left\lfloor \frac{f}{k} \right\rfloor + 1.$$

Consequently, the maximum achievable censorship resistance score is

$$\text{CR} \;=\; \frac{1}{T} \;\le\; \frac{1}{\left\lfloor \frac{f}{k} \right\rfloor + 1} \;\le\; \frac{1}{\frac{f}{k} + 1} \;=\; \frac{k}{f+k},$$

whenever $k \le f$.

Moreover, whenever $k > f$, the transaction τ is assigned to at least one honest proposer in every round, so the adversary cannot censor it even for a single round and $C_{\max} = 1$. Combining the two cases yields

$$\mathsf{CR} \leq \begin{cases} \dfrac{k}{f+k}, & \text{if } k \leq f, \\ 1, & \text{if } k > f, \end{cases}$$

as claimed. □

3.2 MirBFT

As discussed earlier, MirBFT can be viewed as a special instance of $\Pi_{k\text{-dup}}$ with a duplication factor of $k = 1$. Note that MirBFT is among the few BFT systems that explicitly specify a transaction-assignment mechanism as part of the design, rather than relying on the proposers to pick their own transactions freely.

From Theorem 2 and Theorem 3, the throughput and censorship resistance scores of MirBFT are given by $\frac{n-f}{n}$ and $\frac{1}{f+1}$, respectively. MirBFT achieves the highest possible throughput (since each transaction is assigned to exactly one proposer) but exhibits the weakest censorship resistance, as a transaction assigned to a faulty proposer may experience the maximum delay of $O(f)$ rounds. Consequently, MirBFT represents one extreme of the design spectrum – optimal for clients that prioritize throughput over censorship resistance.

3.3 Full Duplication

At the opposite end of the design spectrum lies the Full Duplication variant, corresponding to an instantiation of $\Pi_{k\text{-dup}}$ with $k = f+1$. In this setting, every transaction is assigned to $f+1$ distinct proposers, ensuring that at least one honest proposer proposes it, regardless of which f proposers are Byzantine. This redundancy guarantees achieving the optimal censorship resistance score of 1.

However, this improvement in censorship resistance comes at the cost of throughput. Since each transaction consumes $f+1$ proposer slots per round, the total number of unique transactions that can be proposed is reduced by a factor of $f+1$. From Theorem 2, the throughput in this regime is therefore $\frac{n-f}{n(f+1)} = O(1/f)$, which represents the lowest possible throughput among proposer non-reactive assignment protocols. Consequently, the Full Duplication scheme exemplifies the opposite extreme from MirBFT: it provides perfect censorship resistance but at the expense of minimal throughput.

4 Lower Bounds

In this section, we present our lower bound statements for deterministic, proposer-non-reactive protocols. First, we prove that any deterministic assignment protocol achieving constant censorship resistance score must incur throughput score $\mathsf{THR} = O(1/f)$; we prove this for transaction-non-reactive (Theorem 4) and transaction-reactive (Theorem 5) protocols. Conversely, any deterministic protocol that achieves near-optimal throughput score must tolerate a censorship resistance score $\mathsf{CR} = O(1/f)$; we prove this for transaction-non-reactive (Theorem 6) and transaction-reactive (Theorem 7) protocols.

Throughout the section, we assume that the number of corrupted proposers is a constant fraction of the total number of proposers, i.e., $\frac{f}{n} = \eta$. Also, recall from the validity conditions (Section 2.4) that the maximum duplication factor increase in any round is $\frac{n}{f} = 1/\eta$.

Theorem 4 (Constant Censorship vs. Throughput for all-non-reactive assignment)**.** *Consider a deterministic, proposer-non-reactive,* ***transaction-non-reactive*** *assignment protocol* Π*. Suppose that* Π *has a censorship resistance score of* CR *and throughput score is* THR*. If* $\mathsf{CR} = \frac{1}{k}$ *where* k *is a fixed constant, then the throughput score satisfies* $\mathsf{THR} = O(1/f)$*, where* f *is the number of byzantine faults.*

Proof. If a protocol has censorship resistance score of CR, then by Definition 6 the delay value D_Π for that protocol is $\frac{1}{\mathsf{CR}} = k$, i.e., the protocol must commit any transaction tx assigned first in any round r by round $r+k$. Consider a transaction tx first assigned to a proposer in some round r. Define $\mathcal{P}(\mathsf{tx}, i)$ to be the set of proposers assigned to the transaction tx in round $i \geq r$. Then, we claim that $|\bigcup_r^{r+k} \mathcal{P}(\mathsf{tx}, i)| \geq f+1$. Otherwise, i.e., $|\bigcup_r^{r+k} \mathcal{P}(\mathsf{tx}, i)| \leq f$, then the adversary can choose to corrupt this particular set of proposers, resulting in tx not being committed by round $r+k$. Since the assignment is *proposer-non-reactive* and *transaction-non-reactive*, the duplication factor is independent of the list of proposers $\mathcal{P}$ and the transaction history Hist. Furthermore, as the assignment is balanced (validity condition 4), the duplication factor is also not a function of other transactions, and we can conclude that in each round the duplication factor is just dependent on the round number, i.e., $d(\mathsf{tx}, i) = d_i = |\mathcal{P}(\mathsf{tx}, i)|$, and thus $\sum d_i \geq \left|\bigcup \mathcal{P}(\mathsf{tx}, i)\right|$. Hence,

$$\forall \mathsf{tx} \in A_r : \sum_{i=r}^{r+k} d(\mathsf{tx}, i) \geq f+1 \implies \sum_{i=r}^{r+k} d_i \geq f+1$$

This implies that in at least one round r^*, the duplication factor $d_{r^*} \geq \frac{f+1}{k}$. From the validity conditions, the maximum duplication factor increase in any round is $\frac{n}{f} = 1/\eta$. Thus, in round r, the duplication factor must be

$$d_r \geq \frac{f+1}{k} \cdot \left(\frac{f}{n}\right)^{r^*-r} = \frac{(f+1)\eta^{r^*-r}}{k} \geq \frac{(f+1)\eta^{k-1}}{k} = \Omega(f) \quad (1)$$

By the balanced validity condition 4, any new transaction must have the same duplication factor, and thus, for all rounds, the duplication factor must be $\Omega(f)$.

Consider the throughput under the strategy that the adversary chooses to behave honestly, i.e., broadcast its assigned transactions ($\mathcal{A}$ = honest). The number of transactions committed if all parties broadcast in round i is given by $\frac{nB}{d_i}$, where d_i is $\Omega(f)$. So the average number of committed transactions in any window is given by

$$T_\Pi^{\mathcal{A}=\text{honest}} = \frac{1}{D}\sum_{i=r}^{r+D} \frac{nB}{d_i} \leq \frac{k}{D\eta^{k-1}} \sum_{i=r}^{r+D} \frac{nB}{f+1} = \frac{knB}{(f+1)\eta^{k-1}}$$

Since $T_\Pi \leq T_\Pi^{\mathcal{A}=\text{honest}}$, i.e., any other adversarial strategy can only reduce T_Π (Definition 8) Thus, $\mathsf{THR}_\Pi = \frac{T_\Pi}{nB} \leq \frac{k}{(f+1)\eta^{k-1}} = O(1/f)$. □

Theorem 5 (Constant Censorship vs. Throughput for a transaction-reactive but proposer-non-reactive assignment)**.** *Consider a deterministic, proposer-non-reactive, but* ***transaction-reactive*** *assignment protocol* Π*. Suppose that, when all honest proposers follow* Π*, the censorship resistance score is* CR *and the throughput score is*

THR. *If* $\mathsf{CR} = \frac{1}{k}$ *where* k *is a fixed constant, then the throughput satisfies* $\mathsf{THR} = O(1/f)$.

Proof. The proof for the theorem follows the same steps as Theorem 4. All duplication factors in addition to r are also dependent on Hist, which would not change any step of the proof. □

Corollary 1. *Consider a deterministic, proposer-non-reactive assignment protocol* Π. *Suppose that* Π *has a censorship resistance score of* CR *and the throughput score is* THR. *If* $\mathsf{CR} = 1$, *then the throughput satisfies* $\mathsf{THR} \leq \frac{1}{f+1}$.

Proof. Substituting $k = 1$ in equation 1 implies that the duplication factor for each round r must be $d_r \geq f + 1$. Thus, in any window W, the average number of committed transactions (if all adversarial parties act honest) would be $T_\Pi^{\mathcal{A}=\text{honest}} = \frac{nB}{d_r} \leq \frac{nB}{f+1}$. Thus, $\mathsf{THR}_\Pi = \frac{T_\Pi}{nB} \leq \frac{1}{f+1}$. □

Theorem 6 (Optimal Throughput vs. Censorship for transaction-non-reactive assignment)**.** *Consider a deterministic, proposer non-reactive,* ***transaction-non-reactive*** *assignment protocol* Π. *Suppose that, when all honest proposers follow* Π, *the censorship resistance score is* CR *and the throughput score is* THR. *If* $\mathsf{THR} = \frac{n-f}{n}$, *then the censorship resistance score satisfies* $\mathsf{CR} \leq 1/f$.

Before proving Theorem 6, we state and prove Lemma 2 about the duplication factor of assignment protocols achieving optimal throughput.

Lemma 2. *Let* Π *be a deterministic, proposer-non-reactive,* ***transaction non-reactive*** *assignment protocol with optimal throughput* $\mathsf{THR}_\Pi = \frac{n-f}{n}$. *Then for any* $w \in \mathbb{N}$, *where* $f \leq w << s$, *there exists sufficiently large* $s \in \mathbb{N}$ *such that for all execution traces, and for all* $\mathcal{A} \in \mathcal{A}^*$ $\mathsf{Tr}_\Pi^{\mathcal{A}}(s) = \{M_r, A_r, C_r\}_{r\in[s]}$, *there exists* $r^* < s - w$ *such that* $d(\mathsf{tx}, r) = 1$ *for all rounds* $r \in [r^*, r^* + w]$ *and all transactions* $\mathsf{tx} \in \cup_{r\in[r^*,r^*+w]} A_r$.

Proof. We give a proof by contradiction. Assume that there exists w such that for all sufficiently large $s \in \mathbb{N}$, there exists an execution trace $\mathsf{Tr}_\Pi^{\mathcal{A}}(s) = \{M_r, A_r, C_r\}_{r\in[s]}$ such that for all $r < s - w$, there exists a round $r^* \in [r, r + w]$ and a transaction $\mathsf{tx}^* \in A_{r^*}$, such that $d(\mathsf{tx}^*, r^*) \geq 2$.

Fix any sufficiently large $s \in \mathbb{N}$. We partition the (assumed) trace $\mathsf{Tr}_\Pi^{\mathcal{A}}(s)$ with the above property into w-sized sub-traces, where the j-th sub-trace is $\{M_r, A_r, C_r\}_{jw\leq r<(j+1)w}$. Each sub-trace j must contain at least one a round $r_j \in [j \cdot w, (j + 1) \cdot w]$ and one transaction $\mathsf{tx}_j \in A_{r_j}$, such that $d(\mathsf{tx}_j, r_j) \geq 2$.

Fix one such sub-trace j. Since Π is **transaction-non-reactive**, and using the **balanced** condition, we have $d(\mathsf{tx}, r_j) \geq 2$ for all $\mathsf{tx} \in A_{r_j}$. This means that under the same $\mathcal{A}$ (from $\mathsf{Tr}_\Pi^{\mathcal{A}}(s)$), the number of transactions committed in round r_j is $|C_{r_j}| \leq \frac{n-f}{n}|A_{r_j}| \leq \frac{n-f}{n}\frac{nB}{2}$. Thus the average throughput over sub-trace j is

$$\frac{1}{w}\sum_{r\in[jw,(j+1)w]} |C_r| \leq \frac{1}{w}\Big(\frac{n-f}{n}\frac{nB}{2} + (w-1)\frac{n-f}{n}(nB)\Big)$$
$$= \frac{1}{w}\frac{n-f}{n}(nB)(w - \frac{1}{2}) < \frac{n-f}{n}(nB).$$

Since this is true for all sub-traces within the s-sized trace, and since s is sufficiently large, we get that $T_\Pi^{\mathcal{A}^*} < \frac{n-f}{n}(nB)$, per Definition 7. Therefore the throughput score of Π is $T_\Pi^{\mathcal{A}} < \frac{n-f}{n}$, a contradiction. □

Proof of Theorem 6. Let Π be any assignment protocol with $\mathsf{THR} = \frac{n-f}{n}$. We describe an adversarial strategy $\mathcal{A}$ which causes $D_\Pi^{\mathcal{A}} \geq f$.

Adversarial Strategy $\mathcal{A}$. Given Π, the adversary first examines the protocol description to find a window w with $w \geq f$ such that the duplication factor satisfies $d(\mathsf{tx}, r) = 1$ for every round $r \in [r^*, r^* + w]$ and every transaction $\mathsf{tx} \in \bigcup_{r\in[r^*,r^*+w]} A_r$. The existence of such a window is guaranteed by Lemma 2. Consequently, the adversary executes the following strategy:

(1) Chooses $\mathsf{tx}^* := \mathsf{tx}_1 \in M_{r^*}$.
(2) Initializes $\mathcal{F}^* := \emptyset$.
(3) For $r \in [r^*, r^* + f]$: Simulates $A_r \leftarrow \Pi(\mathcal{P}, r, M_r, \mathsf{Hist}_r)$, Updates $\mathcal{F}^* \leftarrow \mathcal{F}^* \cup A_r[\mathsf{tx}^*]$.
(4) Outputs $\mathcal{F}^*$.

Under adversary $\mathcal{A}$, we observe that tx^* is always assigned to corrupted proposers for the first f rounds of the window w of Π's execution. This means $D^{\mathcal{F}^*} \geq f$ and consequently $D \geq f$. It remains to show that $\mathcal{F}^* \leq f$. By Lemma 2 we know that $d(\mathsf{tx}, r) = 1$ for all $\mathsf{tx} \in M_r$ and all rounds $r \in [f]$, and thus $|A_r[\mathsf{tx}^*]| \leq 1$ for all rounds $r \in [f]$. Therefore, in each iteration of $\mathcal{A}^*$'s simulation loop, the size of $\mathcal{F}$ increases by at most 1, completing the proof. □

Theorem 7 (Optimal Throughput vs. Censorship for transaction-reactive assignment)**.** *Consider a deterministic, proposer-non-reactive,* ***transaction-reactive*** *assignment protocol* Π. *Suppose that, when all honest proposers follow* Π, *the censorship resistance score is* CR *and the throughput score is* THR. *If* $\mathsf{THR} = \frac{n-f}{n}$, *then the censorship resistance score satisfies* $\mathsf{CR} \leq 1/f$.

Proof. Let us assume to the contrary; that the censorship resistance score for such a protocol is $\Pi \geq \frac{1}{f-1}$, i.e., the delay under all adversaries $D_\Pi \leq f - 1$. Consider any transaction tx^r that is first assigned in round r. Since $D_\Pi \leq f - 1$, the transaction tx^r must be committed by round $r + f - 1$ for any adversary. This implies that the duplication factor for the transaction tx in some round $r^* \in [r, r + f - 1]$ is ≥ 2 (otherwise a corruption set that includes each party that is assigned this transaction exists).

Consider an adversary $\mathcal{A} = \text{RANDOM}$, which chooses its corruption set randomly. The probability that the transaction tx^r is censored until round r^*, and then assigned to two honest parties in round r^* is given by

$$\mathbb{P} = \left(\prod_{l=0}^{r^*-r-1} \frac{f-l}{n-l}\right) \cdot \frac{\binom{n-f}{2}}{\binom{n-(r^*-r)}{2}}$$
$$\geq \left(\prod_{l=0}^{f-2} \frac{f-l}{n-l}\right) \cdot \frac{\binom{n-f}{2}}{\binom{n-f+1}{2}} = \frac{f!(n-f+1)!}{n!} \cdot \frac{n-f-1}{n-f+1}.$$

In a sufficiently large but finite window s, the number of such transactions would be at least one, thus reducing the throughput

score strictly from the maximum score of $\frac{n-f}{n}$. Thus, a contradiction. □

5 Improved Assignment Protocols

5.1 Transaction-Reactive (k, ℓ)-Duplication

Next, we study transaction-reactive (k, ℓ)-duplicate assignment protocols as an improvement over transaction-non-reactive protocols (see Section 3.1). Here, in every round, each party is assigned a block of transactions. Each transaction initially has a duplication factor of k; that is, it is assigned to k different parties. If a transaction fails to be committed, its duplication factor is increased additively or multiplicatively by a predefined factor ℓ and it is reassigned in the next round only to parties that it has not previously been assigned to (we assume duplicates are assigned in ascending party-index order whenever the protocol prescribes an ordered placement). Specifically, we focus on a subclass of these protocols in which the duplication factor of a transaction increases over time if the transaction remains uncommitted. Intuitively, if a transaction with an initial duplication factor k is censored in a given round, the protocol prioritizes it in subsequent rounds by assigning it to additional parties, thereby increasing its chances of commitment (i.e., reducing the chances of censorship). We consider two variants of this mechanism: additive and multiplicative duplication. In the additive variant, which we call $\Pi_{(k,\ell)\text{-AI}}$ (Algorithm 3), the duplication factor increases by a constant ℓ in each round, such that the duplication factor in round r is $k_r = k + (r-1) \cdot \ell$. Importantly, the newly assigned parties in each round are distinct from those in previous rounds—for instance, if the transaction was assigned to k_1 parties in round 1, then in round 2 it is assigned to a $k_1 + \ell$ new parties. In the multiplicative variant, which we call $\Pi_{(k,\ell)\text{-MI}}$ (Algorithm 4), the duplication factor grows by a constant multiplicative rate l. Thus, if a transaction remains uncommitted for r rounds, its duplication factor in round r becomes $k_r = \ell^{(r-1)} \cdot k_1$. In this section, we focus on analyzing the worst-case adversarial attack that minimizes throughput and maximizes censorship resistance. We derive a general expression for the throughput and censorship resistance scores as a function of the number of malicious parties f and the duplication factors k and ℓ.

Censorship Resistance. We first analyze the worst-case adversarial strategy for censoring a transaction and derive the resulting censorship resistance score. In the worst case, the adversary targets one transaction, and corrupts all proposers to which that transaction is assigned, until its total corruption budget of f parties is exhausted. Let k_r denote the duplication factor of the targeted transaction in round r. The maximum censorship delay D is characterized by the largest number of rounds for which the adversary can corrupt all assigned proposers, i.e., $\sum_{r=1}^{D} k_r \le f < \sum_{r=1}^{D+1} k_r$. The censorship resistance score is then given by $\mathsf{CR} = 1/(D+1)$. The following lemma states the censorship resistance scores for $\Pi_{(k,\ell)\text{-MI}}$ and $\Pi_{(k,\ell)\text{-AI}}$.

Theorem 8. *The censorship resistance score of the additive-increase protocol* $\Pi_{(k,\ell)\text{-AI}}$ *is* $\mathsf{CR} = \frac{2\ell}{-(2k-3\ell)+\sqrt{(2k-\ell)^2+8\ell f}}$, *while the censorship resistance score of the multiplicative-increase protocol* $\Pi_{(k,\ell)\text{-MI}}$ *is* $\mathsf{CR} = \frac{1}{1+\log_\ell\left(1+\frac{(\ell-1)f}{k}\right)}$.

Algorithm 3 (k, ℓ)-Duplication Additive-Increase Assignment

function $\Pi_{(k,\ell)\text{-AI}}(\mathcal{P}, r, M, \mathsf{Hist}, B)$
  ▷ *Initialize an empty mapping A_r from transactions to sets of proposers*
  $A_r := \emptyset$
  AvailableProposers $\leftarrow \mathcal{P}$
  Sort M by increasing $\mathsf{Hist}[\mathsf{tx}]$ ▷ *Previously assigned transactions (smaller* $\mathsf{Hist}[\mathsf{tx}]$*) have higher priority*
  for each transaction $\mathsf{tx} \in M$ in sorted order **do**
    $r_{\mathsf{tx}} \leftarrow \mathsf{Hist}[\mathsf{tx}]$ ▷ *Round index when* tx *was first assigned to any proposer*
    $d \leftarrow r' - r_{\mathsf{tx}}$ ▷ *Number of rounds since the first assignment of* tx
    $k_{\mathsf{tx}} \leftarrow \min(n,\ k + d \cdot \ell)$ ▷ *Current duplication factor*
    $P_{\mathsf{tx}} \leftarrow$ proposers who have *never* been assigned tx
    $S_{\mathsf{tx}} \leftarrow$ first k_{tx} proposers from P_{tx}
    for each proposer $p \in S_{\mathsf{tx}}$ **do**
      if block of p in $\mathcal{A}_r$ has $< B$ transactions **then**
        Add tx to block of proposer p
      if block of p is now full (B transactions) **then**
        Remove p from AvailableProposers
        if AvailableProposers $= \emptyset$ **then**
          break; ▷ *Stop if all blocks filled*
  return $\mathcal{A}_r$

Algorithm 4 (k, ℓ)-Duplication Multiplicative-Increase Assignment

function $\Pi_{(k,\ell)\text{-MI}}(\mathcal{P}, r, M, \mathsf{Hist}, B)$
  ▷ *Initialize an empty mapping A_r from transactions to sets of proposers*
  $A_r := \emptyset$
  AvailableProposers $\leftarrow \mathcal{P}$
  Sort M by increasing $\mathsf{Hist}[\mathsf{tx}]$ ▷ *Previously assigned transactions (smaller* $\mathsf{Hist}[\mathsf{tx}]$*) have higher priority*
  for each transaction $\mathsf{tx} \in M$ in sorted order **do**
    $r_{\mathsf{tx}} \leftarrow \mathsf{Hist}[\mathsf{tx}]$ ▷ *Round index when* tx *was first assigned to any proposer*
    $d \leftarrow r' - r_{\mathsf{tx}}$ ▷ *Number of rounds since the first assignment of* tx
    $k_{\mathsf{tx}} \leftarrow \min(n,\ k \cdot \ell^d)$ ▷ *Current duplication factor*
    $P_{\mathsf{tx}} \leftarrow$ AvailableProposers who have *never* been assigned tx
    $S_{\mathsf{tx}} \leftarrow$ first k_{tx} proposers from P_{tx}
    for each proposer $p \in S_{\mathsf{tx}}$ **do**
      if block of p in $\mathcal{A}_r$ has $< B$ transactions **then**
        Add tx to block of proposer p
      if block of p is now full (B transactions) **then**
        Remove p from AvailableProposers
        if AvailableProposers $= \emptyset$ **then**
          break; ▷ *Break out of both loops if all blocks filled*
  return $\mathcal{A}_r$

Proof. *1. Additive increase.* For an additive duplication increase factor l, the number of newly assigned parties grows linearly each round, reducing the problem to an arithmetic series. The total number of corrupted parties after D is

$$f = \sum_{r=1}^{D} (k + (r-1)\ell) = Dk + \frac{D(D-1)}{2}\ell$$
$$2f = 2Dk + D(D-1)\ell, = \ell D^2 + (2k-\ell)D.$$

Solving the quadratic equation $\ell D^2 + (2k-\ell)D - 2f = 0$ for D yields

$$D = \frac{-(2k-\ell) + \sqrt{(2k-\ell)^2 + 8\ell f}}{2\ell}$$

This gives the maximum number of rounds D that the adversary can delay the transaction's commitment under additive growth. From Definition 6, the censorship resistance score of Algorithm 3 is

$$\mathsf{CR}_{\text{add}} = \frac{1}{D+1} = \frac{2\ell}{-(2k-3\ell) + \sqrt{(2k-\ell)^2 + 8\ell f}}$$

2. Multiplicative increase. Assume first that the duplication increase factor satisfies $\ell > 1$. In round r, the transaction is assigned to $k\ell^{r-1}$ parties. If the adversary wishes to censor the transaction for D consecutive rounds, it must corrupt all of these parties in each round, so the total number of corrupted parties satisfies

$$f = \sum_{r=1}^{D} k\ell^{r-1} = k\frac{\ell^D - 1}{\ell - 1}.$$

Solving for D yields

$$\ell^D = 1 + \frac{(\ell-1)f}{k} \quad \Rightarrow \quad D = \log_\ell\left(1 + \frac{(\ell-1)f}{k}\right).$$

By Definition 6, the censorship resistance score of Algorithm 4 is therefore

$$\mathsf{CR}_{\Pi} = \frac{1}{D+1} = \frac{1}{\log_\ell\left(1 + \frac{(\ell-1)f}{k}\right) + 1},$$

For the special case $\ell = 1$: the transaction is assigned to k new parties in each round. In this regime the total number of corrupted parties needed to censor the transaction for D rounds is $f = kD$, so $D = f/k + 1$ and hence $\mathsf{CR}_{\Pi} = k/(f+k)$. This reduces to the transaction non-reactive assignment protocol(see Section 3.1). □

This shows that the censorship resistance of the MI assignment protocol is asymptotically better (in terms of f) than that of the AI assignment protocol.

Throughput. As defined in Section 2.1, throughput is measured in the steady state, once the duplication process reaches a stable pattern. Because assignments are transaction-reactive, delaying a transaction by additional rounds strictly increases its number of honest assignees in future rounds, but doing so consumes Byzantine proposer-slots. Thus, throughput is governed by how efficiently the adversary can trade its control over f proposers per round for redundancy among honest proposers in the steady state.

The following lemma states the throughput scores for $\Pi_{(k,\ell)\text{-MI}}$ and $\Pi_{(k,\ell)\text{-AI}}$.

Theorem 9. *Assume the adversary can map duplicated transactions to a disjoint set of honest parties. Then $\Pi_{(k,\ell)\text{-MI}}$ has throughput score $\mathsf{THR}_{\Pi_{(k,\ell)\text{-MI}}} \geq \frac{\max(n-\ell f,\, k)}{nk}$, and $\Pi_{(k,\ell)\text{-AI}}$ of Algorithm 3 has throughput score $\mathsf{THR}_{\Pi_{(k,\ell)\text{-AI}}} \geq \frac{\max(n-\frac{k+\ell}{k}f,\, k)}{nk}$.*

Proof. Consider a window of rounds of size s, across which the throughput score would be evaluated (with $s \to \infty$), per Definition 8. Let m represent the total number of unique transactions committed during the window. During this window, the adversary has control over $s \cdot f \cdot B$ transaction slots, whereas honest parties have $s \cdot (n-f) \cdot B$ slots. To compute the throughput, we compare the cost the adversary pays in its own slots to increase redundancy in the honest slots.

Consider an arbitrary transaction tx assigned to at least one honest proposer in round r, and let $\mathcal{P}(\mathsf{tx}, i)$ represent the set of proposers assigned tx in round $i \in [a, r]$, where a and r the first and last round it got assigned, respectively. Under our transaction-reactive assignment rule, these sets are disjoint across rounds: $\mathcal{P}(\mathsf{tx}, i) \cap \mathcal{P}(\mathsf{tx}, j) = \emptyset$ for $i \neq j$.

If the adversary keeps tx uncommitted through round r, every proposer in $\mathcal{P}(\mathsf{tx}, i)$ for each round $i < r$ must be Byzantine. Therefore, to censor tx for t rounds, the adversary must "spend" at least

$$\sum_{j=0}^{t-1} |\mathcal{P}(\mathsf{tx}, a+j)|$$

Byzantine proposer-slots. Each additional honest copy of tx beyond the first reduces the number of unique committed transactions. If there are $k(\mathsf{tx})$ copies for transaction tx, the adversary reduces the unique count by $k(\mathsf{tx}) - 1$. Thus, we define redundancy introduced by each transaction as $R(\mathsf{tx})$. Note here that the sum of unique transactions, m, and the redundancy introduced by each transaction $\sum_{\mathsf{tx}\in m}(R(\mathsf{tx}))$ is $(n-f)sB$, since all the honest proposer slots must be filled by either a unique transaction or a redundant transaction.

$$m + R(\mathsf{tx}) = (n-f)sB$$

Also, note that in any round the honest parties will commit at least one transaction, and thus across s rounds, at least s transactions will be committed for both MI and AI. This implies that $m \geq s$.

1. Multiplicative Increase. For MI, a transaction of age d has $k_d = k\ell^d$ assignees. To reach age d, the adversary uses $\sum_{i=0}^{d-1} k_i = k\frac{\ell^d - 1}{\ell - 1}$ slots. Given the total corrupted-node cost $\sum_{j=1}^{m} k\frac{\ell^{d_j}-1}{\ell-1} \leq sfB$, it follows that:

$$\sum_{j=1}^{m} k(\ell^{d_j} - 1) \leq (\ell-1)sfB$$

The adversary seeks to maximize redundancy $\Sigma = \sum_{j=1}^{m}(k_{d_j} - 1)$:

$$\Sigma = \sum_{j=1}^{m}(k\ell^{d_j} - k + k - 1) \leq (\ell-1)sfB + (k-1)m$$

With $m = s(n-f)B - \Sigma$ unique transactions:

$$m \geq s(n-f)B - ((\ell-1)sfB + (k-1)m)$$
$$km \geq sB(n-\ell f) \implies m \geq \frac{sB}{k}(n-\ell f)$$

Since $m \geq s \implies m \geq \max\left(s, \frac{sB}{k}(n-\ell f)\right)$, Definition 7 gives

$$T^{\mathcal{A}}_{\Pi_{(k,\ell)\text{-MI}}} := \lim_{s\to\infty}\left(\frac{1}{s}\sum_{r=1}^{s}\left|C_r\right|\right).$$

Under all adversaries $\mathcal{A}$, $\sum_{r=1}^{s}\left|C_r\right| = m \geq \max\left(s, \frac{sB}{k}(n-\ell f)\right)$. Thus,

$$T_{\Pi_{(k,\ell)\text{-MI}}} \geq \lim_{s\to\infty}\left(\frac{1}{s}\max\left(s, \frac{sB}{k}(n-\ell f)\right)\right|).$$

From Definition 8,

$$\text{THR}_{\Pi_{(k,\ell)\text{-MI}}} = \frac{T_{\Pi_{(k,\ell)\text{-MI}}}}{nB} \geq \frac{\max\left(k, (n-\ell f)\right)}{nk},$$

2. Additive Increase. For AI, a transaction of age d has $k_d = k + \ell d$ assignees. To reach age d (i.e., being censored for d rounds), the total Byzantine proposer-slots the adversary must spend is the sum of assignees in all preceding rounds:

$$A(d) = \sum_{j=0}^{d-1}(k + j\ell) = kd + \frac{\ell d(d-1)}{2} \leq sfB.$$

Whenever tx is finally committed at age d, it occupies k_d slots. One slot is the unique transaction, and the remaining $k_d - 1$ slots are redundant copies. Thus, the redundancy created by a transaction of age d is $R(d) = k + d\ell - 1$.

We observe that for any $d \geq 0$, the redundancy is bounded by:

$$R(d) \leq \frac{\ell}{k}A(d) + (k-1).$$

The adversary chooses a multiset of ages $\mathcal{D} = \{d_j\}_{j=1}^{m}$ to maximize total redundancy $\Sigma = \sum_{j=1}^{m} k + d_j\ell - 1$ subject to the total Byzantine slot constraint $\sum_{j=1}^{m}\left(kd_j + \frac{\ell d_j(d_j-1)}{2}\right) \leq sfB$. Using our linear bound:

$$\begin{aligned}\Sigma = \sum_{j=1}^{m} R(d_j) &\leq \sum_{j=1}^{m}\left(\frac{\ell}{k}A(d_j) + k - 1\right)\\ &\leq \frac{\ell}{k}sfB + m(k-1).\end{aligned}$$

The number of unique committed transactions m is the difference between total honest slots and the redundancy Σ:

$$\begin{aligned}m &= s(n-f)B - \Sigma\\ m &\geq s(n-f)B - \left(\frac{\ell}{k}sfB + m(k-1)\right)\\ km &\geq sB\left(n - \left(1 + \frac{\ell}{k}\right)f\right)\\ m &\geq \frac{sB}{k}\left(n - \frac{k+\ell}{k}f\right).\end{aligned}$$

Since $m \geq s \implies m \geq \max\left(s, \frac{sB}{k}\left(n - \frac{k+\ell}{k}f\right)\right)$, Definition 7 gives

$$T^{\mathcal{A}}_{\Pi_{(k,\ell)\text{-AI}}} := \lim_{s\to\infty}\left(\frac{1}{s}\sum_{r=1}^{s}\left|C_r\right|\right).$$

Under all adversaries $\mathcal{A}$, $\sum_{r=1}^{s}\left|C_r\right| = m \geq \max\left(s, \frac{sB}{k}\left(n - \frac{k+\ell}{k}f\right)\right)$. Thus,

$$T_{\Pi_{(k,\ell)\text{-AI}}} \geq \lim_{s\to\infty}\left(\frac{1}{s}\max\left(s, \frac{sB}{k}\left(n - \frac{k+\ell}{k}f\right)\right)\right).$$

From Definition 8,

$$\text{THR}_{\Pi_{(k,\ell)\text{-AI}}} = \frac{T_{\Pi_{(k,\ell)\text{-AI}}}}{nB} \geq \frac{\max\left(k, \left(n - \frac{k+\ell}{k}f\right)\right)}{nk}$$

□

On AI vs. MI. By Theorems 8 and 9, the throughput of both additive-increase (AI) and multiplicative-increase (MI) assignment protocols is a linear function of the adversarial corruption budget f. For fixed duplication parameters (k, ℓ) we obtain

$$T_{\Pi_{(k,\ell)\text{-MI}}} = B\left(\frac{n-\ell f}{k}\right) \quad \text{and} \quad T_{\Pi_{(k,\ell)\text{-AI}}} = \frac{B\left(n - \left(\frac{k+\ell}{k}\right)f\right)}{k}.$$

Thus, comparing worst-case throughput reduces to comparing the coefficients of f. We have

$$T_{\Pi_{(k,\ell)\text{-MI}}} \geq T_{\Pi_{(k,\ell)\text{-AI}}} \iff \ell \leq \left(\frac{k+\ell}{k}\right) \iff \ell(k-1) \leq k.$$

Since ℓ is usually small (~ 2), the throughput for AI and MI do not differ substantially. However, as shown in Theorem 8, the two models differ sharply in censorship resistance: multiplicative increase achieves censorship resistance of order $1/\log_\ell(f/k)$, while additive increase achieves only order of $\sqrt{\ell/f}$. Consequently, multiplicative-increase assignment protocols offer stronger censorship resistance while maintaining throughput behavior that is comparable (up to constant factors) to that of additive-increase protocols.

5.2 Transaction-Reactive (k, ℓ)-Duplication vs. Transaction-Non-Reactive k-Duplication

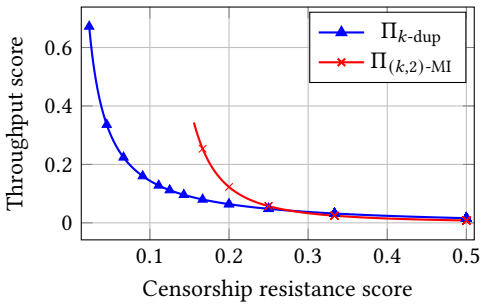

Figure 2: Throughput vs. censorship resistance score for $n = 128$, $f = 42$, $\ell = 2$, with $k \in [1, f]$.

To compare the transaction-reactive protocol $\Pi_{(k,\ell)\text{-MI}}$ against the transaction non-reactive protocol $\Pi_{k\text{-dup}}$, we plot in Figure 2 the score tuples (CR, THR) attained by each protocol family. Specifically, we compare $\Pi_{(k,2)\text{-MI}}$ against $\Pi_{k\text{-dup}}$ in the same regime with $n = 128$ proposers out of which $f = \lfloor \frac{1}{3} \cdot 128 \rfloor = 42$ are Byzantine. For each possible CR value, we compute the k value achieving CR and then use k to compute the corresponding THR value.

Figure 2 shows that $\Pi_{(k,\ell)\text{-MI}}$ outperforms $\Pi_{k\text{-dup}}$ in most of the region where $\Pi_{(k,2)\text{-MI}}$ is defined, i.e., for THR $\leq 1/3$. Concretely, except for the extreme case where the duplication factor is 1, any throughput score that $\Pi_{k\text{-dup}}$ can achieve, $\Pi_{(k,\ell)\text{-MI}}$ can achieve the same throughput score with a much (exponentially) better censorship resistance score. As an example, Sui, which is essentially $\Pi_{5\text{-dup}}$,

is strictly dominated by $\Pi_{(1,2)\text{-MI}}$, which gives better throughput and censorship resistance scores.

6 Randomized Assignment Protocols

In this section, we study randomized assignment protocols, obtained by randomizing the k-duplication assignment family from Section 3.1. Recall that the deterministic round-robin assignment is predictable and can be targeted: an adversary can censor a transaction by corrupting its k designated proposers. Our randomized variants instead sample assignees uniformly at random (avoiding repeat assignments across rounds), and we assume an adversary that chooses its per-round strategy before learning that round's randomness.

We present two assignment protocols: $\Pi^{\text{tx}}_{k\text{-rand}}$ and $\Pi^{\text{proposer}}_{k\text{-rand}}$. In $\Pi^{\text{tx}}_{k\text{-rand}}$, each *transaction* is randomly assigned to exactly k proposers, so that each proposer is assigned B transactions in expectation. Note that this scheme requires *global* randomness. In $\Pi^{\text{proposer}}_{k\text{-rand}}$, each *proposer* randomly samples B transactions such that each transaction is assigned to k distinct proposers in expectation. This scheme can be instead instantiated with *local* randomness sampled by each proposer.

Modified Throughput Score and Censorship Resistance Score Definitions. In a randomized protocol, if we take an infinite interval, some transaction's delay can be infinitely long with a small probability. Similarly, it is possible that all honest parties select the same transactions.

On the throughput side, we redefine the throughput score to be based on *expected* throughput. For throughput under adversary (Definition 7) and throughput (Definition 8), we use the same definitions for randomized protocols; the only difference is that throughput is now measured in expectation, provided that we can prove tight concentration bounds around the expected value.

On the censorship resistance side, we redefine the censorship resistance score to be based on worst-case censorship resistance *except with a small probability* parameterized by parameter c.

Definition 10 (c−Delay Under Adversary)**.** *Let Π be a randomized assignment protocol and $\mathcal{A}$ a fixed adversary strategy. Let $\{M_r, A_r, C_r\}_{r=1}^{\infty}$ be the (random) execution trace. For $s \in \mathbb{N}$, let $\mathcal{T}_s' := \bigcup_{r=1}^{s} A_r$. For each $\mathsf{tx} \in \mathcal{T}_s'$, define $D(\mathsf{tx}) := c_{\mathsf{tx}} - a_{\mathsf{tx}}$, where c_{tx} and a_{tx} are the first round that tx was assigned and committed respectively. For $\alpha \in (0,1)$, define the α-*percentile delay in s rounds as*

$$Q_\alpha(s) \;:=\; D \quad \textit{s.t.}\ \frac{1}{|\mathcal{T}_s'|} \cdot \mathbb{E}\left(\left|\{\mathsf{tx} \in \mathcal{T}_s' : D(\mathsf{tx}) \le D\}\right|\right) \;\ge\; \alpha.$$

*We define the c−*censorship delay *of Π under $\mathcal{A}$ to be*

$$D_\Pi^{\mathcal{A}} \;:=\; \lim_{s\to\infty} Q_\alpha(s) \quad \textit{for } \alpha := 1 - (nB)^{-c}.$$

Definition 11 (c−Censorship Resistance)**.** *The c−censorship resistance score is defined as in Definition 6, but with c−censorship delay for adversaries (from Definition 10).*

6.1 Sampling k Proposers per Transaction

We describe the protocol $\Pi^{\text{tx}}_{k\text{-rand}}$, formally specified in Algorithm 5. The main idea is that in each round, all proposers share some global randomness, which determines the k proposers each transaction is assigned to. In more detail, in each round, transactions in the mempool are ordered, giving priority to the oldest assigned transactions. For each transaction tx, the protocol samples k proposers uniformly at random from those who have not previously been assigned to tx. In both Algorithms 5 and 6, SampleDistinct(S, k, rand) denotes sampling k distinct elements from set S using random seed rand. Note that $\Pi^{\text{tx}}_{k\text{-rand}}$ can produce assignments that do not explicitly respect the block size B, i.e., there might be proposers who are assigned less or more than B transactions; however, the *expected* block size for each proposer is B. Next, we analyze the throughput and censorship resistance behavior of $\Pi^{\text{tx}}_{k\text{-rand}}$.

Algorithm 5 Per Transaction Sampling

```
function Π^TX_{k-RAND}(P, r, M, Hist, rand)
    ▷ Initialize an empty mapping A_r from transactions to sets of proposers
    A_r := ∅
    AvailableProposers ← P
    Sort M by increasing Hist[tx] ▷ Previously assigned transactions (smaller Hist[tx]) have higher priority
    for each transaction tx ∈ M in sorted order do
        P_tx ← proposers who have never been assigned tx before
        ▷ Sample k distinct proposers uniformly at random from P_tx
        S_tx ← SampleDistinct(P_tx, k, rand)
        for each proposer P ∈ S_tx do
            if A_r[P] has < B transactions then
                Add tx to A_r[P]
                Record that P has now been assigned tx
            if A_r[P] is now full (B transactions) then
                Remove P from AvailableProposers
                if AvailableProposers = ∅ then
                    break;                ▷ Stop if all blocks filled
    return A_r
```

Throughput. In Theorem 10, we prove that the expected throughput score for $\Pi^{\text{tx}}_{k\text{-rand}}$ is $\mathbb{E}[\mathsf{THR}] = \frac{1}{k}\left(1 - \binom{f}{k}/\binom{n}{k}\right)$; in Corollary 2 we simplify and show that $\mathbb{E}[\mathsf{THR}] \ge \frac{1-\eta^k}{k}$, where η is the fault ratio. Under $\Pi^{\text{tx}}_{k\text{-rand}}$, the sampling process is independent across transactions; thus, we can prove (see Lemma 3) that THR is tightly concentrated around its expected value with high probability.

Theorem 10 (Expected Throughput in $\Pi^{\text{tx}}_{k\text{-rand}}$)**.** *The throughput score of assignment protocol $\Pi^{tx}_{k\text{-}rand}$ (described in Algorithm 5) satisfies $\mathbb{E}[\mathsf{THR}] = \frac{p}{k}$, where $p = 1 - \binom{f}{k}/\binom{n}{k}$.*

Proof. Let $|H| = n - f$ denote the number of honest proposers, and Let $T = \frac{n \cdot B}{k}$ be the total number of distinct transactions to be assigned in a given round. Let X be the random variable that denotes the number of transactions assigned to at least one honest proposer. For a fixed transaction, the probability that all k proposers assigned to it are corrupted is

$$\Pr[\text{all } k \text{ are corrupt}] = \frac{\binom{f}{k}}{\binom{n}{k}}.$$

Hence, the probability that the transaction is assigned to at least one honest party is

$$p = 1 - \frac{\binom{f}{k}}{\binom{n}{k}}.$$

By linearity of expectation,

$$\mathbb{E}[X] = T \cdot p = \frac{nB}{k}\left(1 - \frac{\binom{f}{k}}{\binom{n}{k}}\right).$$

Under $\Pi^{\text{tx}}_{k\text{-rand}}$, each round has the same expected throughput. Thus the throughput score per Definition 8 is equal to the expected throughput score for a single round:

$$\mathbb{E}[\mathsf{THR}] \;=\; \frac{\mathbb{E}[X]}{nB} = \frac{p}{k} = \frac{1}{k}\left(1 - \frac{\binom{f}{k}}{\binom{n}{k}}\right).$$

□

Corollary 2 (Expected Throughput lower bound)**.** *Assume $f = \eta \cdot n$ for some constant η. Then $\mathbb{E}[\mathsf{THR}] \geq \frac{1-\eta^k}{k}$.*

Proof. Using that $\frac{\binom{f}{k}}{\binom{n}{k}} < \left(\frac{f}{n}\right)^k$, we get that $p \geq 1 - \left(\frac{f}{n}\right)^k$ and thus

$$\mathbb{E}[\mathsf{THR}] \geq \frac{1}{k}\left(1 - \left(\frac{f}{n}\right)^k\right).$$

Next, using $f = \eta \cdot n$, we can simplify

$$\mathbb{E}[\mathsf{THR}] \geq \frac{1-\eta^k}{k}.$$

□

Lemma 3 (Throughput concentration bound in $\Pi^{\text{tx}}_{k\text{-rand}}$)**.** *Let X denote the number of unique transactions assigned to at least one honest party in a round, and let $\mathsf{THR} = X/(nB)$. Then, for any $0 < \delta < 1$,*

$$\Pr\left[\mathsf{THR} \leq (1-\delta)\,\mathbb{E}[\mathsf{THR}]\right] \leq \exp\left(-\frac{\delta^2\,\mathbb{E}[X]}{2}\right),$$

and similarly

$$\Pr\left[\mathsf{THR} \geq (1+\delta)\,\mathbb{E}[\mathsf{THR}]\right] \leq \exp\left(-\frac{\delta^2\,\mathbb{E}[X]}{3}\right).$$

In particular, for sufficiently large nB/k, the throughput is tightly concentrated around its expectation with high probability.

Proof. Let I_i be the indicator that transaction i is assigned to at least one honest party. Then $X = \sum_{i=1}^{T} I_i$, where $T = nB/k$, and each I_i is an independent Bernoulli random variable with mean p. The result follows directly from standard Chernoff bounds applied to the sum of independent Bernoulli random variables, then dividing by nB to get the throughput score. □

Censorship Resistance. In Theorem 11, we derive the c-censorship resistance score of $\Pi^{\text{tx}}_{k\text{-rand}}$. In Corollary 3, we simplify $\mathsf{CR} \geq \frac{k\ln(1/\eta)}{c\ln(nB)}$; this means that it suffices to set $k \in \Theta(\ln n)$ for CR to equal 1.

Before deriving the censorship resistance score CR in Theorem 11, we first state and prove the censorship delay D in Lemma 4.

Lemma 4 (Censorship delay in $\Pi^{\text{tx}}_{k\text{-rand}}$)**.** *Fix $c \in \mathbb{R}^+$. Under assignment protocol $\Pi^{tx}_{k\text{-}rand}$, the censorship delay of any given transaction is at most*

$$D \geq \frac{c \cdot \ln(nB)}{\ln\left(\frac{\binom{n}{k}}{\binom{f}{k}}\right)}$$

rounds, except with probability at most $\frac{1}{(nB)^c}$.

Proof. Fix a transaction tx. In any given round, tx is censored if all of its assigned proposers are Byzantine, which occurs with probability

$$p_{\text{censor}} = \frac{\binom{f}{k}}{\binom{n}{k}}.$$

Since sampling is independent across rounds, the probability that tx is censored for D rounds is p^D_{censor}. We require that this probability p^D_{censor} is at most $\frac{1}{(nB)^c}$. Rearranging, we get that any given transaction is censored for at most

$$D = \frac{c \cdot \ln(nB)}{\ln\left(\frac{\binom{n}{k}}{\binom{f}{k}}\right)}$$

rounds, except with probability at most $\frac{1}{(nB)^c}$.

□

Theorem 11 (Censorship resistance score of $\Pi^{\text{tx}}_{k\text{-rand}}$)**.** *Fix $c \in \mathbb{R}^+$. The assignment protocol $\Pi^{proposer}_{k\text{-}rand}$ has c-censorship resistance score*

$$\mathsf{CR} \geq \frac{\ln\left(\binom{n}{k} / \binom{f}{k}\right)}{c \cdot \ln(nB)}.$$

Proof. Lemma 4 guarantees that any given transaction is censored for at most

$$D = \frac{c \cdot \ln(nB)}{\ln\left(\frac{\binom{n}{k}}{\binom{f}{k}}\right)}$$

rounds, except with probability at most $\frac{1}{(nB)^c}$. Per Definition 11, this implies that $\Pi^{\text{tx}}_{k\text{-rand}}$ has c-censorship resistance score

$$\mathsf{CR} \geq \frac{\ln\left(\frac{\binom{n}{k}}{\binom{f}{k}}\right)}{c \cdot \ln(nB)}.$$

□

Corollary 3 (Censorship resistance lower bound for $\Pi^{\text{tx}}_{k\text{-rand}}$)**.** *Let $f = \eta n$ for $0 < \eta < 1$. Then the c-censorship resistance score of $\Pi^{tx}_{k\text{-}rand}$ satisfies*

$$\mathsf{CR} \geq \frac{k\ln(1/\eta)}{c\ln(nB)},$$

except with probability at most $\frac{1}{(nB)^c}$.

Proof. From Theorem 11 and using that $\frac{\binom{f}{k}}{\binom{n}{k}} < \left(\frac{f}{n}\right)^k$, we get that

$$\begin{aligned}\mathsf{CR} &\geq \frac{\ln\left(\frac{\binom{n}{k}}{\binom{f}{k}}\right)}{c\ln(nB)} \geq \frac{\ln\left(\frac{1}{(f/n)^k}\right)}{c\ln(nB)} \\ &= \frac{k\ln(1/\eta)}{c\ln(nB)},\end{aligned}$$

as desired. □

6.2 Sampling B Transactions per Proposer

Next, we describe the second randomized assignment protocol $\Pi^{\text{proposer}}_{k\text{-rand}}$, formally specified in Algorithm 6. In each round, and for each proposer P_i, the protocol samples B transactions uniformly at random from the candidate pool of $\frac{nB}{k}$ transactions. Under this sampling method, each transaction has a probability k/n to be selected by any proposer, and thus, an overall (expected) duplication factor of k. We highlight that $\Pi^{\text{proposer}}_{k\text{-rand}}$ can be instantiated with *local* randomness, i.e., each proposer can locally sample randomness, which determines the B transactions to include. Next, we analyze the throughput and censorship behavior of $\Pi^{\text{proposer}}_{k\text{-rand}}$.

Algorithm 6 Per Proposer Sampling

```
1:  function Π^PROPOSER_k-RAND(𝒫, r, M, Hist, rand)
2:      A_r := ∅
3:      AvailableProposers ← 𝒫
4:      Sort M by increasing Hist[tx]
5:      ▷ Define the target candidate pool size based on system capacity
6:      N_cap ← (n · B)/k
7:      CandidatePool ← First N_cap transactions of M
8:      for each proposer P ∈ 𝒫 do
9:          ▷ Uniformly sample exactly B transactions from the candidate pool
10:         A_r[P] ← SampleDistinct(CandidatePool, B, rand)
11:     return A_r
```

Throughput. In Theorem 12, we prove that the expected throughput score for $\Pi^{\text{proposer}}_{k\text{-rand}}$ is $\mathbb{E}[\mathsf{THR}] = \frac{1}{k}\left(1 - \left(1 - \frac{k}{n}\right)^{n-f}\right)$; in Corollary 4 we simplify $\mathbb{E}[\mathsf{THR}] \approx \frac{1-e^{-k(1-\eta)}}{k}$, where η is the fault ratio.

Theorem 12 (Expected Throughput in $\Pi^{\text{proposer}}_{k\text{-rand}}$)**.** *The throughput score of assignment protocol $\Pi^{tx}_{k\text{-rand}}$ (described in Algorithm 6) satisfies $\mathbb{E}[\mathsf{THR}] = \frac{p}{k}$, where $p = 1 - \left(1 - \frac{k}{n}\right)^{n-f}$.*

Proof. Since each proposer samples transactions from the pool of nB/k transactions uniformly at random, the probability that a given transaction is not chosen by a given honest proposer is

$$p_s = 1 - \frac{B}{nB/k} = 1 - \frac{k}{n}.$$

Since each honest proposer samples independently of other proposers, the probability that a given transaction is not chosen by *any* honest proposer is

$$p_a = \left(1 - \frac{k}{n}\right)^{n-f}.$$

Thus, the probability that at least one honest proposer chooses a given transaction is

$$p = 1 - p_a = 1 - \left(1 - \frac{k}{n}\right)^{n-f}.$$

Since there exist $\frac{nB}{k}$ transactions in the candidate pool, the expected throughput is given by

$$\mathbb{E}(X) = \frac{nB}{k} \cdot p,$$

and the throughput score is given by

$$\mathbb{E}(\mathsf{THR}) = \frac{\mathbb{E}(X)}{nB} = \frac{p}{k}.$$

□

Corollary 4. *Let $f = \eta n$ for $0 < \eta < 1$. If n is large, then $\mathbb{E}(\mathsf{THR}) = \frac{1-e^{-k(1-\eta)}}{k}$.*

Proof. We assume that $f = \eta n$. Thus,

$$p = 1 - \left(1 - \frac{k}{n}\right)^{n(1-\eta)} = 1 - e^{-k(1-\eta)}$$

for large n. Thus, the expected throughput score tends to $\frac{1-e^{-k(1-\eta)}}{k}$. □

Next, we want to prove a tail bound on the throughput of $\Pi^{\text{proposer}}_{k\text{-rand}}$, similarly to the analysis of $\Pi^{\text{tx}}_{k\text{-rand}}$. However, note that the indicator random variables X_i – that represent whether the i-th transaction is proposed by at least one honest proposer – are *not* independent under $\Pi^{\text{proposer}}_{k\text{-rand}}$. In order to use Chernoff bounds, we prove that the random variables are *negatively associated (NA)* (Lemma 5). After establishing NA, we can use Chernoff–Hoeffding bounds (Lemma 6) to get tail bounds for the throughput of the protocol. Our proof relies on (Proposition 5, [12]) which states that Chernoff–Hoeffding bounds are applicable.

Lemma 5 (Negative Association of Transaction Inclusion)**.** *Let X_i be an indicator random variable representing whether a particular transaction tx_i is included by a at least one honest proposer, under $\Pi^{proposer}_{k\text{-rand}}$. The set of variables $\{X_i\}_{\forall i}$ are negatively associated (NA).*

Proof. To prove this statement we will prove first that for each proposer, selecting a transaction is NA to selecting other transactions and then through the standard closure properties defined by [17] extend the negative association to the at least one honest proposer case.

Let $X_{i,j}$ be a random variable that honest proposer j selects transaction tx_i. Restating the problem as: An urn (mempool) contains $N(= nB/k)$ balls (transactions), each having a different color (unique transactions). Suppose a random sample of $n(= k)$ balls (transactions) is chosen (without replacement) and $Y_i, i = 1, \ldots, N$, be random variables indicating the presence of a ball of the ith color in the sample. This proves the equivalence to the example

stated in Section 3.1(c) of [17] which proves that this distribution is Negatively Associated.

From Property P_7 of [17], we have that a union of independent sets of NA variables is NA. Since each honest proposer chooses its set independent of each other, the union of chosen transactions from each proposer results in a Negative Association between the transactions. □

Now that we have established Negative Association, we will use Chernoff–Hoeffding bounds to get tail bounds for the throughput of the system.

Lemma 6 (Throughput concentration bound for $\Pi^{\text{proposer}}_{k\text{-rand}}$)**.** *Let X be the random variable representing the number of unique transactions committed by honest proposers in a single round. Let $\mu = \mathbb{E}[X]$ be the expected throughput derived in Theorem 12. For any $\delta \in (0,1)$, the probability that the realized throughput deviates from the expectation is bounded by:*

$$\Pr[X \le (1-\delta)\mu] \le \exp\left(-\frac{\delta^2\mu}{2}\right)$$

and

$$\Pr[X \ge (1+\delta)\mu] \le \exp\left(-\frac{\delta^2\mu}{3}\right)$$

Proof. Let X_i be the random variable that transaction i is assigned to at least one honest party. Then $X = \sum_{i=1}^{T} X_i$, where $T = nB/k$, and each X_i are negatively associated random variable with mean p. The result follows directly from standard Chernoff bounds applied to the sum of negatively associated random variables, then dividing by nB to get the throughput score. □

Censorship Resistance. In Theorem 13, we derive the c-censorship resistance score of $\Pi^{\text{proposer}}_{k\text{-rand}}$. In Corollary 5, we show that $\mathsf{CR} \ge \frac{k(1-\eta)}{c\cdot\ln(nB)}$; similar to $\Pi^{\text{tx}}_{k\text{-rand}}$, it suffices to set $k \in \Theta(\ln n)$ for CR to equal 1.

Before deriving the censorship resistance score CR in Theorem 13, we first state and prove the censorship delay D in Lemma 7.

Lemma 7 (Censorship delay in $\Pi^{\text{proposer}}_{k\text{-rand}}$)**.** *Fix $c \in \mathbb{R}^+$. Under assignment protocol $\Pi^{\textit{proposer}}_{k\text{-rand}}$, the c-censorship delay of any given transaction is at most*

$$D = \frac{n\cdot c\cdot \ln(nB)}{k(n-f)}$$

rounds, except with probability at most $\frac{1}{(nB)^c}$.

Proof. Fix a transaction tx and an honest proposer P. The probability that tx is not proposed by P in one round is

$$1 - \frac{B}{nB/k} = 1 - \frac{k}{n}.$$

Since honest proposers select transactions to propose independently of other proposers, and using that $(1-x)^{n-f} \le e^{-x(n-f)}$, the probability that tx is not proposed by *any* honest proposer (in one round) is given by

$$\left(1-\frac{k}{n}\right)^{n-f} \le e^{-k(n-f)/n}.$$

Over D independent rounds, the probability that tx is never proposed by any honest proposer is at most

$$(e^{-k(n-f)/n})^D = e^{-Dk(n-f)/n}.$$

We require that this probability is at most $\frac{1}{(nB)^c}$. Rearranging, we get that any given transaction is censored for at most

$$D = \frac{n\cdot c\cdot \ln(nB)}{k(n-f)}$$

rounds, except with probability at most $\frac{1}{(nB)^c}$.

□

Theorem 13 (Censorship resistance score of $\Pi^{\text{tx}}_{k\text{-rand}}$)**.** *Fix $c \in \mathbb{R}^+$. The assignment protocol $\Pi^{\textit{proposer}}_{k\text{-rand}}$ has c-censorship resistance score*

$$\mathsf{CR} \ge \frac{k(n-f)}{n\cdot c\cdot \ln(nB)}.$$

Proof. Lemma 7 guarantees that any given transaction is censored for at most

$$D = \frac{n\cdot c\cdot \ln(nB)}{k(n-f)}$$

rounds, except with probability at most $\frac{1}{(nB)^c}$. Per Definition 11, this implies that $\Pi^{\text{proposer}}_{k\text{-rand}}$ has c-censorship resistance score

$$\mathsf{CR} \ge \frac{k(n-f)}{n\cdot c\cdot \ln(nB)}.$$

□

Corollary 5 (Censorship resistance lower bound for $\Pi^{\text{proposer}}_{k\text{-rand}}$)**.** *Let $f = \eta n$ for $0 < \eta < 1$. Then $\Pi^{\textit{proposer}}_{k\text{-rand}}$ has c-censorship resistance score*

$$\mathsf{CR} \ge \frac{k(1-\eta)}{c\cdot \ln(nB)}.$$

Proof. Plugging in $\eta = f/n$ in Theorem 13, we get that $\Pi^{\text{proposer}}_{k\text{-rand}}$ has c-censorship resistance score

$$\mathsf{CR} \ge \frac{k(1-\eta)}{c\cdot \ln(nB)}.$$

□

6.3 Comparison Between Randomized and Deterministic k-Duplicate Protocols

To showcase the power of adding randomness in an assignment protocol, we compare the two *randomized* k-duplicate protocols with the *deterministic* k-duplicate transaction-non-reactive protocol.

Comparison Setting. We plot the THR and CR scores against the duplication factor k for the three protocols. To facilitate comparison, we set $\eta = \frac{1}{3}$ and evaluate the THR and CR formulas with $n = 128$ proposers out of which $f = \lfloor \frac{1}{3}\cdot 128 \rfloor = 42$ proposers are Byzantine. For CR, we set the block size to be $B = 5$ and consider the $(c = 2)$-censorship resistance score for both randomized protocols.

Comparison Observations. In Figures 3 and 4, we plot the THR and CR scores against the duplication factor $k \in [1, 10]$. In Figure 3, we observe that the randomized protocols consistently achieve better throughput than the deterministic counterpart. The only exception occurs when $k = 1$, where $\Pi^{\text{proposer}}_{1\text{-rand}}$ fails to achieve the maximum $\mathsf{THR} = 1 - \eta = 2/3$ that is achieved by both $\Pi^{\text{tx}}_{1\text{-rand}}$ and

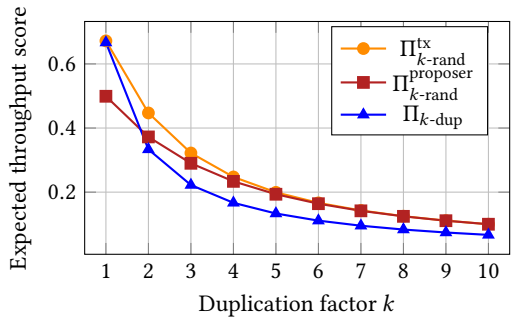

Figure 3: Expected throughput score vs. duplication factor of deterministic and randomized k-duplicate protocols. Here $n = 128$, $\eta = 1/3$, and $f = 42$.

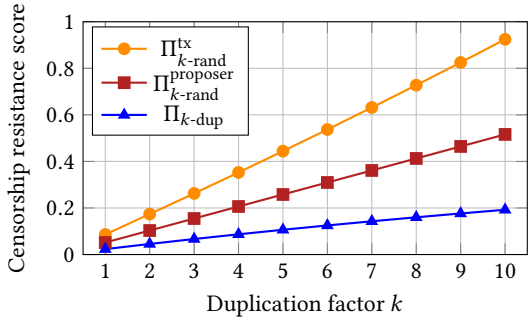

Figure 4: Censorship resistance score vs. duplication factor k of deterministic and randomized k-duplicate protocols. Here $n = 128$, $\eta = 1/3$, $f = 42$, $B = 5$, and $c = 2$.

$\Pi_{1\text{-dup}}$. In $\Pi^{\text{proposer}}_{k\text{-rand}}$, honest proposers sample transactions independently; thus THR reflects that some transactions receive no proposer and are missed, and this is why THR is below the $1 - \eta$ baseline (for $k = 1$). On the other hand, $\Pi^{\text{tx}}_{k\text{-rand}}$ is better than $\Pi^{\text{proposer}}_{k\text{-rand}}$ because it assigns each transaction to exactly k proposers, preventing "unassigned" transactions.

In Figure 4, we observe that the randomized protocols *significantly outperform* the deterministic counterpart in terms of censorship resistance. In this $n = 128$ regime, $\Pi^{\text{tx}}_{k\text{-rand}}$ consistently achieves more than 4x higher CR than $\Pi_{k\text{-dup}}$. We emphasize that this difference grows *exponentially* in the number of proposers n. As in the THR case, $\Pi^{\text{tx}}_{k\text{-rand}}$ achieves better CR scores than $\Pi^{\text{proposer}}_{k\text{-rand}}$. In $\Pi^{\text{proposer}}_{k\text{-rand}}$, duplication is only guaranteed in expectation, yielding weaker worst-case censorship resistance; in contrast, $\Pi^{\text{tx}}_{k\text{-rand}}$ assigns each transaction to exactly k proposers, eliminating under-assigned transactions and thus achieving higher censorship resistance.

6.4 Overcoming Deterministic Lower Bounds

Our randomized protocols overcome deterministic lower bounds. Here, we discuss how $\Pi^{\text{tx}}_{k\text{-rand}}$ overcomes both "classes" of deterministic lower bounds from Section 4, as also illustrated in Figure 1.

First, if we set $k^* = \frac{c \ln(nB)}{\ln(1/\eta)} \ll f$, then $\Pi^{\text{tx}}_{k^*\text{-rand}}$ achieves c-censorship resistance score of 1, while $\text{THR} \in \Theta(\frac{\ln(n)}{n})$. Therefore, $\Pi^{\text{tx}}_{k^*\text{-rand}}$ overcomes the lower bound from Theorem 5, which states that protocols achieving optimal $\text{CR} = 1$ must suffer from $\text{THR} \in O(1/f)$.

Second, $\Pi^{\text{tx}}_{1\text{-rand}}$ achieves $\text{THR} = 1 - \eta$, where $\eta = f/n$, and c-censorship resistance score $\text{CR} \geq \frac{\ln(1/\eta)}{c \cdot \ln(nB)}$. Therefore, $\Pi^{\text{tx}}_{1\text{-rand}}$ overcomes the lower bound from Theorem 6, which states that protocols achieving $\text{THR} = 1 - \eta$ must suffer from $\text{CR} \leq \frac{1}{f}$.

7 Related Work

Traditional BFT consensus [8, 9, 27] relies on a single leader per view, creating a single point of censorship. The adversary can censor a transaction for f blocks in cases where a byzantine leader's silence triggers a view change protocol to elect a new leader. While the overall throughput of such systems remains optimal (since no duplicate transactions are inserted), the absolute value remains small, since a single leader can only propose a certain number of transactions capped by its maximum bandwidth.

Multi-proposer (leaders) consensus protocols aim to address this limitation. A dominant approach is DAG-based consensus, which decouples transaction dissemination from ordering. Mir-BFT [24] introduced parallel leaders with deterministic transaction partitioning to improve throughput. DAG-based consensus protocols, such as Bullshark [23], Narwhal and Tusk [11], and protocols such as Aptos and Sui allow every validator to propose blocks concurrently [4, 6]. If a client submits a transaction to sufficiently many validators (e.g., more than $2f$), at least one honest proposer will include it, yielding strong short-term censorship resistance. In the worst case, insufficient dissemination (only sent to constant number of proposers) yields an $O(f)$ censorship delay, while sufficiently wide dissemination yields constant delay. However, the choice of sending the transaction to multiple proposers reduces the throughput of the system, since multiple parties would propose the same transaction, and thus at best a fraction $O(1/f)$ of the maximum throughput would be achieved as informed by our lower bound results in Section 4.

To reduce the bandwidth overhead induced by transaction duplication, recent work such as Sedna proposes a dissemination model in which users transmit erasure-coded fragments rather than full transactions [21]. This significantly reduces the bandwidth cost of reaching multiple proposers, but introduces additional practical constraints. In particular, fragment verification does not enforce correct encoding: a proposer can only check that a fragment is well-formed and signed, not that it corresponds to a valid or decodable transaction. Our work focuses on the full-transaction dissemination regime instead, which is the dominant model used in practice.

Another related work [2] measures the minimum latency cost that a transaction must face to ensure censorship resistance, and shows how to achieve this bound. Such a lower bound is still subject to our results. Thus, to ensure censorship resistance one must pay a cost in latency for finality as well as in throughput.

Outside of pure consensus protocol design, one line of work augments a single-proposer model with inclusion lists (FOCIL [25] and AUCIL [26]) – additional constraints that force a block proposer to include certain transactions. In FOCIL, for each slot a small

committee of validators is randomly chosen to each gossip a local inclusion list of transactions. The block proposer must then include an aggregate of these lists in its block, and attesters check that this aggregate honestly reflects the committee's reports. AUCIL refine this approach by introducing a fee based assignment of transaction to the committee members through a correlated equilibrium based coordination scheme. Multiple IL proposers independently select transactions, and an aggregator is chosen via an auction that rewards larger inclusion lists. Excluding transactions is therefore disincentivized, since an aggregator that omits transactions risks losing to one that includes more. Both FOCIL and AUCIL strengthen censorship resistance in single BFTs – either through committee vetos or economic incentives, but introduce additional overhead from committee communication and inclusion-list data, and do not mitigate transaction duplications.

8 Conclusion and Future Work

This work presents a unified framework for analyzing the trade-off between censorship resistance and throughput in multi-proposer BFT protocols. We formalize quantitative metrics for censorship resistance and throughput, establish tight lower bounds for deterministic protocols, and construct deterministic and randomized schemes that match or exceed these bounds. An interesting direction for future work is to analyze these trade-offs under rational adversarial models, where proposers are utility-maximizing agents. In such settings, it becomes necessary to account for strategic behavior, incentive compatibility, and equilibrium outcomes, potentially yielding different assignment mechanisms and feasible regions. Finally, it would be interesting to study the same censorship-vs-throughput tradeoff problem under *weighted* settings. For instance, proposers might control more building power (e.g., stake), or transactions might consume different compute (e.g., gas used).

References

[1] 0xJessica. 2024. Braid: Enhancing Ethereum's Censorship Resistance Through Technological Innovation. https://www.gate.io/learn/articles/braid-enhancing-ethereums-censorship-resistance-through-technological-innovation/4130

[2] Ittai Abraham, Yuval Efron, and Ling Ren. 2025. The Latency Cost Of Censorship Resistance. Cryptology ePrint Archive, Paper 2025/2136. https://eprint.iacr.org/2025/2136

[3] Orestis Alpos, Bernardo David, Nikolas Kamarinakis, and Dionysis Zindros. 2024. *Flashbots Report: System Requirements, Existing and New Solutions, and Their Efficiency.* Technical Report. Flashbots.

[4] Aptos Foundation. 2025. Aptos White Paper. https://aptos.dev/network/blockchain/aptos-white-paper. Accessed 2025-11.

[5] Ava Labs. 2020. Avalanche Consensus Protocol. https://docs.avax.network/overview/avalanche-consensus.

[6] Kushal Babel, Andrey Chursin, George Danezis, Lefteris Kokoris-Kogias, and Alberto Sonnino. 2024. Mysticeti: Low-Latency DAG Consensus with Fast Commit Path. doi:10.48550/arXiv.2310.14821 arXiv:2310.14821 [cs].

[7] Juan Benet et al. 2017. Filecoin: A Decentralized Storage Network. https://filecoin.io/filecoin.pdf.

[8] Ethan Buchman, Jae Kwon, and Zarko Milosevic. 2019. The latest gossip on BFT consensus. arXiv:1807.04938 [cs.DC] https://arxiv.org/abs/1807.04938

[9] Miguel Castro, Barbara Liskov, et al. 1999. Practical byzantine fault tolerance. In *OsDI*, Vol. 99. 173–186.

[10] CensorshipPics 2025. Censorship.pics. https://censorship.pics. Accessed 2025.

[11] George Danezis, Lefteris Kokoris-Kogias, Alberto Sonnino, and Alexander Spiegelman. 2022. Narwhal and tusk: a dag-based mempool and efficient bft consensus. In *Proceedings of the Seventeenth European Conference on Computer Systems*. 34–50.

[12] Devdatt P. Dubhashi and Desh Ranjan. 1996. Balls and Bins: A Study in Negative Dependence. *BRICS Report Series* 3, 25 (1996), 1–23.

[13] Pranav Garimidi, Joachim Neu, and Max Resnick. 2025. Multiple Concurrent Proposers: Why and How. Cryptology ePrint Archive, Paper 2025/1772. https://eprint.iacr.org/2025/1772

[14] Suyash Gupta, Dakai Kang, Dahlia Malkhi, and Mohammad Sadoghi. 2025. Carry the Tail in Consensus Protocols. *arXiv preprint arXiv:2508.12173* (2025).

[15] Hedera Hashgraph, LLC. 2020. The Hedera Hashgraph Consensus Algorithm. https://hedera.com/hh_whitepaper_v2.1-20200815.pdf.

[16] Mohammad Mussadiq Jalalzai and Kushal Babel. 2025. MonadBFT: Fast, Responsive, Fork-Resistant Streamlined Consensus. *arXiv preprint arXiv:2502.20692* (2025).

[17] Kumar Joag-Dev and Frank Proschan. 1983. Negative association of random variables with applications. *The Annals of Statistics* (1983), 286–295.

[18] Quentin Kniep, Jakub Sliwinski, and Roger Wattenhofer. 2025. *Alpenglow: A Consensus Protocol for a High-Performance Proof of Stake Blockchain.* Technical Report. Anza Technology, Inc. https://www.anza.xyz/blog/alpenglow-a-new-consensus-for-solana White Paper (v1.0, May 19 2025) for Solana upgrade "Alpenglow".

[19] Mysten Labs. 2022. Sui: A platform for high-performance smart contracts. https://github.com/MystenLabs/sui/blob/main/doc/paper/sui.pdf. Accessed 2025-11.

[20] Mysten Labs. 2025. sui. https://github.com/mystenlabs/sui. Accessed: 2025.

[21] Alejandro Ranchal-Pedrosa, Benjamin Marsh, Lefteris Kokoris-Kogias, and Alberto Sonnino. 2025. Sedna: Sharding transactions in multiple concurrent proposer blockchains. *arXiv* abs/2512.17045 (2025).

[22] RedBelly Network. 2019. The RedBelly Network Whitepaper. https://redbelly.network/hubfs/The%20Redbelly%20Network%20Whitepaper%201.4.pdf.

[23] Alexander Spiegelman, Neil Giridharan, Alberto Sonnino, and Lefteris Kokoris-Kogias. 2022. Bullshark: DAG BFT Protocols Made Practical. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22)*. Association for Computing Machinery, New York, NY, USA, 2705–2718. doi:10.1145/3548606.3559361

[24] Chrysoula Stathakopoulou, Tudor David, Matej Pavlovic, and Marko Vukolić. 2021. Mir-BFT: High-Throughput Robust BFT for Decentralized Networks. arXiv:1906.05552 [cs.DC] https://arxiv.org/abs/1906.05552

[25] Thomas Thiery, Barnabe Monnot, Luca Zanolini, and Julian Ma. 2024. Fork-Choice Enforced Inclusion Lists (FOCIL). https://ethresear.ch/t/fork-choice-enforced-inclusion-lists-focil-a-simple-committee-based-inclusion-list-proposal/19870. Ethereum Research post.

[26] Sarisht Wadhwa, Julian Ma, Thomas Thiery, Barnabe Monnot, Luca Zanolini, Fan Zhang, and Kartik Nayak. 2025. AUCIL: An Inclusion List Design for Rational Parties. https://eprint.iacr.org/2025/194 Publication info: Preprint..

[27] Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham. 2019. HotStuff: BFT Consensus with Linearity and Responsiveness *(PODC '19)*. Association for Computing Machinery, New York, NY, USA, 347–356. doi:10.1145/3293611.3331591