© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720196

IsoBFT: A Novel Byzantine Fault-Tolerant Consensus Algorithm for Ultra-Low-Latency Decentralized Networks in Critical Infrastructure and Industrial IoT

HASSAN CESSI IBRAHIM[1], DAMILARE TIMOTHY OGUNJOBI[2], PHILIP MENSAH[3]
[1]*J. W. McClure School of Emerging Communication Technologies, Ohio University, Athens, Ohio, USA*
ORCID: 0009-0009-4527-8472
[2]*Department of Computer Engineering, Federal University of Technology Akure, Ondo State, Nigeria*
ORCID ID - 0009-0007-0861-5092
[3]*University of Minnesota, OIT PeopleSoft Solutions, Minneapolis, Minnesota*
0009-0007-2283-4056

Abstract- ***The networks that run operational technology (OT) substations, water treatment plants, oil and gas pipelines, and manufacturing lines are moving from a centralized control to a federated, multi-stakeholder architecture coordinated by permissioned distributed ledgers. Protection and control loops in the electrical grid and other critical infrastructure have protection-relay tripping times, IEC 61850 GOOSE message classes, and SCADA/PMU polling cycles that impose multi-millisecond to sub-second deadlines on protection and control operations, while Byzantine fault-tolerant (BFT) consensus protocols like PBFT, Tendermint, HotStuff, and HoneyBadgerBFT were designed for settlement workloads that can tolerate hundreds of milliseconds to seconds of latency. In this paper, we survey four representative BFT families, discuss their structural latency and scalability constraints for OT deployment, and introduce a hybrid consensus algorithm called IsoBFT (Isochronous Byzantine Fault Tolerance), which combines an optimistic single-round-trip fast path with a PBFT-style fallback mechanism based on a network-stability monitor, and elects a small rotating committee using a verifiable random function (VRF). A formal system model, safety/liveness/termination proof, and security analysis for eight attack classes are provided, with a proposition quantifying the degradation of the practical availability of the safety guarantee when the global Byzantine fraction is approaching one-third. Using realistic Modbus/DNP3/IEC 61850 OT traffic, the discrete-event simulation of the design IsoBFT managed to execute realistic workloads with median consensus latency ranging from 4.90ms at n = 10-50 to 9.17-11.26ms at n = 100 and n = 500, remaining competitive with or better than PBFT and Tendermint across this range. Committee-bounded communication overhead stayed essentially flat with respect to the number of validators from n = 10 to n = 50, but newly completed runs at n = 100 and n = 500 (n = 200 still outstanding) show overhead growing faster than the quadratic scaling of PBFT and Tendermint over that range, together with a heavy P95/P99 latency tail not present at smaller scale; this discrepancy with the theoretical scale-independence result is reported and discussed rather than resolved. IsoBFT could reduce the median latency by approximately 81% and 56% under up to 33% Byzantine faults compared to HotStuff and HoneyBadgerBFT, respectively, at n = 10-50, while maintaining the safety of the system; a Byzantine-resilience sweep at n = 100 shows a narrower advantage over PBFT/Tendermint than at smaller scale.***

Keywords: ***Byzantine Fault Tolerance, Consensus Algorithms, Critical Infrastructure, Industrial Internet Of Things, Operational Technology.***

I. INTRODUCTION

1.1 The Convergence of Operational Technology and Decentralized Ledgers

Critical infrastructure industries like electric power generation, water and wastewater treatment, oil and gas pipelines, transportation, and advanced manufacturing are transitioning from a more centralized supervisory control environment to a more decentralized and collaborative operational environment. This transformation is spurred by the fast development of Industrial Internet of Things (IIoT) technologies, distributed energy resources, edge computing, and multi-stakeholder industrial ecosystems, in which a number of independent entities need to securely share operational data and synchronously make control decisions.

Permissioned distributed ledger technologies are now a viable platform to enable trusted coordination

between mutually distrustful parties in the absence of a central coordinating authority. These systems can ensure integrity, accountability, and resilience to malicious or faulty participants by using a shared, tamper-evident ledger with Byzantine fault-tolerant (BFT) consensus. As a result, distributed ledgers are now emerging as a potential solution for several applications including distributed energy resource coordination, cross-utility protection systems, collaborative industrial automation, supply-chain traceability and secure machine-to-machine communication.

However, there are challenges in implementing DLT in operational technology environments. Industrial control systems are unlike traditional blockchain applications in that they are subject to real-time requirements were taking a decision too late can put equipment at risk, services out of service, or lives at stake. Thus, enterprise or financial system consensus protocols cannot simply be reused without taking into account the high latency and reliability demands of industrial processes.

1.2 Why Sub-Second Latency Is a Hard OT Requirement

With respect to distributed consensus, existing Byzantine fault-tolerant consensus protocols, such as PBFT, Tendermint, HotStuff, and HoneyBadgerBFT, have made significant progress towards consensus protocols that are more secure, scalable, and fault-tolerant in the face of adversarial network conditions. These protocols, however, are mostly designed for blockchain and distributed ledger applications where the transaction confirmation time is in the order of hundreds of milliseconds to several seconds. Their communication structure includes several phases of voting, leader-election mechanisms, view-change mechanisms, and all-to-all message exchange, which can be a huge communication overhead when the number of validators increases.

These are different from operational technology environments. In industrial communication networks like IEC 61850, Modbus, DNP3, and SCADA networks, protection and control actions must take place within a few milliseconds to ensure system stability and operational safety. The traditional multi-phase consensus approach in protection relays, programmable logic controllers, and distributed control systems will create unacceptably high latency, which may lead to delayed fault isolation, cascading failures, or reduced operational resilience.

Many such suggestions have been proposed, such as optimistic execution, consensus by committee, linear communication protocols, verifiable random function (VRF) based validator selection and DAG based dissemination, but they typically aim to fix a single issue rather than all 4 of the above-mentioned issues. Therefore, no protocol fulfills all the stringent operational requirements of critical infrastructure deployments, which are latency-sensitive.

This is because there are three conflicting objectives: maintaining the Byzantine fault tolerance, minimizing the number of communications as the network size increases, and ensuring low-latency consensus in both good and bad network conditions. There is still an open research problem for a distributed consensus problem that satisfies all these objectives.

1.3 Contributions

In this paper, we introduce a hybrid consensus protocol called IsoBFT (Isochronous Byzantine Fault Tolerance) that is tailored for latency-sensitive operational technology and Industrial IoT environments to overcome the shortcomings of the current Byzantine fault-tolerant consensus protocols. Instead of inventing new consensus primitives, IsoBFT is a systematic combination of existing ones (such as VRF-based committee election, bounded-committee consensus, optimistic single-round execution, proactive network-stability monitoring, and a PBFT-inspired fallback protocol) in a single architecture tailored to real-time industrial communication.

The main innovation of IsoBFT is that it proactively adapts to the network conditions that are being observed. The protocol constantly monitors the stability of the network by measuring round-trip time, jitter, and packet loss and picks the best path to execute the protocol without waiting for communication failures or timeout events. IsoBFT runs a communication-delay-minimizing lightweight optimistic consensus path under favourable network conditions. If instability or Byzantine behaviour is

detected, the protocol seamlessly switches to a bounded PBFT-style recovery mechanism, ensuring safety and liveness while maintaining correctness.

The main outcomes of this work are described below: In-depth review of the current BFT protocols. We systematically study the most representative consensus families, such as PBFT, Tendermint, HotStuff, and HoneyBadgerBFT, and pinpoint the architectural limitations that prevent these from applying to latency-sensitive operational technology and Industrial IoT environments.

Design of the IsoBFT consensus framework. We propose a hybrid Byzantine fault-tolerant consensus protocol which combines VRF-based committee selection, proactive network-aware path selection, optimistic single-round consensus execution, DAG-assisted transaction dissemination, and bounded PBFT-like fallback to achieve low-latency consensus with classical Byzantine fault tolerance.

Formal correctness and security analysis. We define a formal system model and prove safety, liveness, termination, communication complexity, and security properties of IsoBFT under partial synchrony, as well as analyze risks and resilience of committee selection in the face of representative attack scenarios.

Comprehensive performance evaluation. We conduct extensive discrete-event simulations using realistic operational technology communication workloads from Modbus, DNP3, and IEC 61850 environments to evaluate the performance of IsoBFT and compare with existing BFT protocols in terms of consensus latency, throughput, communication overhead, scalability, and resilience to Byzantine faults.

1.4 Paper Organization

The rest of this paper is structured around the standard structure requested for this manuscript. The literature review of related BFT consensus research and formalization of the research gap is Section 2. Section 3 presents the formal system model, IsoBFT design and pseudo-code, formal correctness analysis, security analysis against 8 attack classes, and a simulation-based experimental setup. The outcomes of the simulations, the ablation study, and the sensitivity analysis are displayed in Section 4. Section 5 discusses the risk implications for critical infrastructures, situations where IsoBFT is not applicable, limitations, and threats to validity. The conclusion and future work are provided in Section 6.

II. LITERATURE REVIEW

The Byzantine agreement problem was first introduced by Lamport, Shostak and Pease as the problem of reaching reliable agreement among distributed processes in the presence of arbitrarily faulty (“traitorous”) processes (Lamport et al., 1982). Later, Fischer, Lynch, and Paterson were able to prove that deterministic consensus is impossible in a fully asynchronous system even when only one process can be faulty, which resulted in randomized and timing-based solutions (Fischer et al., 1985). The partial-synchrony model, where the delay of messages is not predictable, is analyzed, and the fast path and fallback of IsoBFT (Dwork et al., 1988) are used. We present four representative algorithms in this section that span the design space of decentralized OT networks: partially-synchronous leader-based (PBFT, Tendermint), linear pipelined (HotStuff), an asynchronous (HoneyBadgerBFT) one, and an extensive survey of related designs, which leads to the identified research gap.

2.1 Practical Byzantine Fault Tolerance (PBFT)

Castro and Liskov's PBFT (Castro & Liskov, 1999) was the first BFT algorithm to be shown to be practical in that it was demonstrated to be able to work at sub-second speed, for small replica sizes, under partial synchrony. It has three phases: pre-prepare, prepare, and commit, in which there are up to f Byzantine replicas out of $n = 3f + 1$ replicas, and a primary replica proposes a value, and all replicas vote using all-to-all authenticated messaging. The upside is that it is a well-studied and well-known safety argument and that it has low latency at small size, while the downside is that it is an “all to all” voting pattern, which scales quadratically with the number of validators, and that view-change (validator replacement after suspected failure) is also an “all to all” sub-protocol, which at the smallest size adds hundreds of milliseconds to seconds of latency, with real world network jitter. Thus, PBFT is a good fit for dense clusters of tens of nodes, but not for the tens to hundreds of validators that are envisioned for multi-utility OT deployments.

2.2 Tendermint

Blockchain-native round-based protocol with accountable safety properties, and a simpler implementation model than PBFT, Tendermint is a consensus engine for blockchain systems that has been designed as a proof-of-stake protocol (Kwon, 2014) and formalized by Buchman. Later adopted by many public and permissioned chains, it still has an essentially quadratic communication pattern with naive gossip, and the design is round-based so that a single slow or unresponsive proposer slows the network down until a timeout, and a new round is started, making it difficult to bound the worst-case and even typical-case latency tightly enough for millisecond-class OT deadlines.

2.3 HotStuff

In HotStuff, the PBFT/Tendermint phase structure is removed, and a three-phase pipelined voting protocol is proposed that only communicates O(n) per view, allowing it and its chained, production-deployed variants to scale to much larger sets of validators without quadratic message growth. However, two more things are still preventing OT applicability: firstly, there are still multiple sequential communication phases (usually three) before finality; secondly, the communication round-trip time is still not in the single-digit millisecond range without reducing the number of validators or changing the commit path, which is not optimized for the bursty, deterministic, poll-driven traffic of OT networks.

2.4 HoneyBadgerBFT

HoneyBadgerBFT (Miller et al., 2016) is the first BFT protocol without any synchronous assumption, and thus it is live even in a fully asynchronously adversarially scheduled network. It offers secure broadcast and asynchronous common subset (ACS) agreement and threshold encryption, which means that no transaction is censored by the batch collector. While the cryptographic and combinatorial machinery it needs is quite a bit slower in the constant factor than partially synchronous protocols, which can operate at sub-second OT control loops, the other end of the latency spectrum, at moderate numbers of validators.

2.5 Other Related Consensus Designs

There were several other designs which had an influence on the construction of IsoBFT. Replica action is speculative in Zyzzyva (Kotla et al., 2007) and takes action on a proposal before the committee has a consensus; conceptually similar, but without a risk of taking a rollback action in IsoBFT as a full committee quorum is required for finality. SBFT (Gueta et al., 2019) uses threshold signatures, optimistic fast path, and random sampling of committees. This work is not a deterministic finality layer, as is Nakamoto's proof-of-work blockchain (Nakamoto, 2008), which only provides probabilistic finality with an order of minutes. Baird's hashgraph protocol (Baird, 2016) is a gossip-about-gossip virtual voting system that is pertinent to IsoBFT's DAG-based propagation layer. There are two proof-of-stake protocols which are directly relevant to IsoBFT's committee election: Algorand (Gilad et al., 2017) privately and randomly selects a small committee per round, and Ouroboros Praos (David et al., 2018) formalizes the VRF-based leader election with a security proof under adaptive corruption, from which IsoBFT's VRF-based election is directly inspired, but tailored to a latency-critical OT setting.

TABLE 1. Comparative summary of BFT consensus algorithms against OT-relevant design properties.

Algorithm	Communication complexity	Synchrony assumption	Typical latency floor	Key OT-relevant limitation
PBFT	$O(n^2)$ per phase	Partial synchrony	Tens of ms (small n) to seconds (large n)	Quadratic blow-up and costly view-change at OT-scale committees
Tendermint	$O(n^2)$ (naive gossip)	Partial synchrony	~1–3 s typical block time in production	Round timeout stalls on a single slow proposer

HotStuff	O(n) per view (linear)	Partial synchrony	3× network RTT (multi-phase pipeline)	Sequential phases impose a latency floor independent of n
HoneyBadgerBFT	$O(n^2)$ w/ threshold crypto	Fully asynchronous	Hundreds of ms to low seconds	High constant-factor cost of ACS + threshold cryptography
IsoBFT (proposed)	$O(k^2)$, $k \ll n$ bounded committee	Partial synchrony w/ optimistic fast path	Single-digit to tens of ms	Fallback path required for adversarial/unstable conditions

2.6 Gaps Specific to IIoT and Critical Infrastructure

Beyond the generic latency and scalability concerns, there are three gaps that are applicable to the deployment of OT and are common across all four of the algorithms reviewed. None was designed to consider the regular, poll- or event-driven traffic characteristics of Modbus, DNP3, and IEC 61850 GOOSE/Sampled Values communications in which many small, time-critical messages are received on deterministic or quasi-deterministic schedules, instead of the bursty, human-driven transactions common in financial and general blockchain workloads. None offers the chance to "take a shortcut" on multi-phase voting when substation- and plant-local networks are often engineered and redundant and impose the same worst-case-oriented latency cost on each phase, irrespective of network health. No one has tried it at the validator counts (tens to low hundreds) that are typical of federated, multi-utility, multi-vendor IIoT deployments, where PBFT's and Tendermint's quadratic costs and even HotStuff's linear but multi-phase costs matter. These three gaps come together to drive the design of IsoBFT.

2.7 Positioning and Novelty of IsoBFT

Some of IsoBFT's features have been explored individually: VRF-based committee election in Algorand (Gilad et al., 2017) and Ouroboros Praos (David et al., 2018); a fast path with a slower fallback quorum in SBFT (Gueta et al., 2019); and network-adaptive protocols that switch between an optimistic synchronous fast path and an asynchronous fallback in Jolteon/Ditto (Danezis et al., 2022) and the Narwhal/Bullshark/Mysticeti family. What is new is not any one mechanism, but the proactive gating logic between them: IsoBFT's network-stability monitor is a proactive, not reactive, input to path selection, based on a rolling estimate of committee-link round-trip time, jitter, and loss, as opposed to Zyzzyva and SBFT, which take a fast-path timeout as a reactive input to path selection, or Jolteon-style designs, which rely on a fixed global synchrony bound. This is a proactive-gating claim which is assessed directly in the ablation study (Section 4.2). By contrast, IsoBFT's VRF election with a fixed (permissioned) set of validators is equivalent to Algorand's sortition, and the k-bounded PBFT-style fallback is equivalent to PBFT with a committee, which is a bounded combination of known mechanisms, not a new one.

The innovation of IsoBFT is that it does not introduce a new consensus primitive, but systematically combines existing ones into a proactive, latency-aware design, particularly for Industrial IoT deployments. Current adaptive BFT implementations only react to degradation in the network, while IsoBFT continually predicts the quality of the network and proactively selects the consensus path before it is affected by the network degradation and has an impact on consensus latency. This design philosophy is different from previous hybrid BFT designs.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720196

TABLE 2. Comparison of IsoBFT with the closest related hybrid/optimistic BFT designs.

Property	Zyzzyva	SBFT	Jolteon-style	Algorand	Narwhal/Bullshark	IsoBFT
Speculative/optimistic fast path	Yes	Yes	Yes	No	No	Yes
Bounded committee ($k \ll n$)	No	Yes	No	Yes (per-block sortition)	No (full validator set)	Yes
VRF-based election	No	No	No	Yes	No	Yes
Proactive stability-gated path selection	No	No	Partial (global Δ only)	N/A	N/A	Yes
DAG-decoupled dissemination	No	No	No	No	Yes	Yes (propagation layer only)
Fallback safety proof restricted to k	N/A	Partial	N/A	N/A	N/A	Yes
Explicit committee-composition risk analysis	No	No	No	Yes (own literature)	N/A	Yes (this paper)

While there are existing studies that have proposed individual techniques like VRF committee selection, optimistic execution, DAG-based dissemination, and adaptive consensus switching, none of them have proposed all these techniques in a single architecture optimized for latency-critical OT environments. Most of the current solutions focus on either scalability or fault tolerance, but not both, when considering deterministic industrial communication constraints. This gap is what spurs the development of IsoBFT.

2.8 Research Gap and Problem Statement

The algorithms reviewed here have a common theme: there is a trade-off between latency and simplicity (PBFT, Tendermint) or worst-case network robustness (HoneyBadgerBFT), or the most latency-friendly classical design, HotStuff, still has a multi-phase commit floor set by the network round-trip time, not the actual, often high, reliability of the underlying links. None of the protocols reviewed consider, in real time, observed network stability as a first-class input to the consensus path itself; all of them assume a worst-case or asynchronous adversary on each round, even though substation and plant-local OT networks are, for the overwhelming majority of the time, stable, low-jitter, and without Byzantine behavior.

The research problem tackled in this paper is the absence of any existing BFT consensus algorithm that simultaneously (a) bounds the worst-case communication complexity independently of the total size of the validator set, (b) achieves single-digit-to-low-tens-of-milliseconds finality under benign network conditions typical to substation and plant OT networks, and (c) provides classical safety and liveness guarantees against up to $f = \lfloor (n-1)/3 \rfloor$ Byzantine validators at the validator-set sizes (tens to low hundreds) envisioned for federated, multi-utility IIoT and critical-infrastructure deployments. IsoBFT is developed and tested to address this.

III. METHODOLOGY

The methodology is structured in four phases: a formal system model, which fixes assumptions on the network, fault, and committee-selection (3.1); the design of the IsoBFT protocol itself, including its consensus flow and pseudocode (3.2); a formal correctness analysis, which proves safety, liveness, termination, and communication-complexity properties (3.3); and a structured security analysis against eight classes of attacks (3.4); and a discrete-event simulation setup to evaluate the protocol empirically (3.5).

3.1 Formal System Model

IsoBFT is a permissioned protocol, which means that the validator set $V = \{1, \ldots, n\}$ is known and fixed and will not change in an epoch. The network is partially synchronous, meaning that after a fixed but unknown Global Stabilization Time (GST), all messages sent by a correct validator at time t are delivered to its correct recipients by $t + \Delta$, for some fixed but unknown Δ, and before GST, message delays are arbitrary and may be adversarially scheduled. It is the same model that PBFT, Tendermint, and HotStuff are traditionally analyzed. The time delay between any two validators at time t is assumed to be one-way and is modelled as belonging to one of two regimes: stable and degraded, with each regime having a mean, a jitter, and a loss probability; this is a modelling choice and not an empirical statement about the behaviour of the network.

The adversary is allowed to corrupt a fixed but unknown set of Byzantine validators $B \subset V$, $|B| = f$, with $f \leq \lfloor (n-1)/3 \rfloor$ (slowly adaptive adversary) for the duration of each epoch, and may change which validators are corrupt between epochs, subject to this budget. The Byzantine validators can refuse to send messages, equivocate, and delay sending messages up to (but not provably more than) the protocol timeout, but cannot forge signatures, invert hashes, or compute another validator's VRF output without the secret key of the other validators.

At the beginning of each epoch e, a committee $C_e \subset V$, $|C_e| = k$, is elected using a verifiable random function: each validator i runs $\text{VRF}(sk_i, seed_e)$ with proof $\pi_{i,e}$ to obtain $y_{i,e}$, where $seed_e$ is a public, unpredictable value based on the hash of the finalized state of epoch e−1 and the construction of Micali, Rabin, and Vadhan (Micali et al., 1999). The two selection rules considered are: (i) threshold (Bernoulli) sortition, as in Algorand (Gilad et al., 2017), where validator i joins C_e if $y_{i,e}$ is below k/n, giving each validator an independent Bernoulli(k/n) chance of selection, and a true committee size distributed as Binomial(n, k/n); and (ii) rank-based sortition, where the validator i is selected iff its VRF output is among the k smallest across V, fixing $|C_e| = k$ at the cost of requiring every validator to learn every other validator's VRF output before the committee is final. The simulator in this paper's evaluation is based on the "rank" rule, while the production design is aimed at Bernoulli sortition.

The number of Byzantine validators actually seated on a particular committee in a given epoch, $X_e = |C_e \cap B|$, is a random variable, which is key to the safety analysis in Section 3.3. Writing $f_k = \lfloor (k-1)/3 \rfloor$ for the number of Byzantine committee members the k-bounded sub-protocols can tolerate, committee safety for epoch e holds iff $X_e \leq f_k$.

Proposition 1 (Committee-Safety Concentration). Let $p = 1/3 - \varepsilon$ for margin $\varepsilon \in (0, 1/3]$ and let C_e be chosen via either of the above sortition rules. Then $\Pr[X_e > f_k] \leq \exp(-2k\varepsilon^2)$. Proof sketch: The tolerance gap $f_k - \mu = k\varepsilon - 1/3$ is positive for $k > 1/(3\varepsilon)$; Hoeffding's inequality for a sum of k bounded (or negatively associated, without-replacement) variables gives the stated bound.

This bound has the following practical implication: both this bound and the actual hypergeometric tail probability (evaluated directly, not using the looser Hoeffding bound) tend to "no guarantee" as p approaches 1/3, for any fixed k, because the gap $f_k - \mu$ approaches zero as $\varepsilon \to 0$. For a committee of $k = 21$ members, the exact probability that more than $f_k = 6$ of them are Byzantine is 58.2%: under the headline 33%-Byzantine test condition, a randomly elected committee is more likely than not to violate the safety precondition assumed by the fast-path and fallback safety theorems below. This does not mean that IsoBFT's design is invalid, but that the claim of resilience needs to be qualified as follows: The per-epoch safety guarantee is unconditional, provided that the committee is safe; the availability of this guarantee, which is the probability that the committee elected during a given epoch is actually safe, drops significantly as the Byzantine fraction approaches 1/3, and is strong only when the Byzantine fraction is bounded away from 1/3 by a margin that must shrink roughly as $1/\sqrt{k}$ as k gets smaller.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720196

TABLE 3. Exact probability that a randomly elected committee violates $X_e \leq f_k$ (n = 100; computed exactly via the hypergeometric distribution).

Global Byzantine fraction p	k = 7	k = 15	k = 21	k = 31	k = 41
10%	2.1%	0.6%	0.06 %	~0%	~0%
20%	14.1 %	14.7 %	8.3%	1.2%	0.4%
30%	35.0 %	48.8 %	44.9 %	28.3 %	29.6 %
33%	42.1 %	59.6 %	58.2 %	44.7 %	50.3 %

Two quorums of the same committee have at least $q(k) - f_k = f_k + 1 \geq 1$ members in common; for a committee of size $k = 3f_k + 1$, a quorum is any subset of size $q(k) = 2f_k + 1$. At last, the network-stability monitor is expressed as a Boolean predicate Stable (C_e, t), which is evaluated locally by each committee member, based on a sliding window of the past w observed RTT samples and loss indicators, and is the conjunction of three threshold conditions on the mean RTT, RTT jitter, and mean loss (θ_RTT, θ_jitter, θ_loss). The fast path is tried if Stable (C_e, t) = 1 at the start of a round; otherwise, the fallback is used, or if the fast path fails to complete due to timeout or equivocation is detected. This formalization is important because it renders the fast path liveness to be explicitly conditional on the predicate, not unconditional.

3.1.2. Network Stability Monitor

The network stability monitor is a runtime component that is embedded within each IsoBFT validator node and is lightweight. Used to determine whether the current communication setting is good enough to use the optimistic fast-path consensus protocol, or whether to switch proactively to the bounded PBFT-style fallback protocol. Monitor does not wait for timeout expiration to determine if the network is stable, but rather continually evaluates recent network activity and makes a determination of stability before every consensus round.

The monitor has a fixed-length sliding observation window of the last (w = 20) round-trip time (RTT) measurements and associated packet-delivery results. Based on these observations, three network quality metrics are calculated: mean RTT, RTT jitter, and packet-loss rate. The communication delay between the members of the committee is measured as the mean RTT, the standard deviation of the samples of RTT is measured as the jitter and the percentage of messages not received within the observation window, among the expected messages, is measured as the packet-loss rate.

When configuring the one-way network delay, the expected round-trip time for stable operating conditions is calculated. Explain the concept of mean one-way delay (\bar{d}). Thus, the round-trip time baseline is:

When configuring the one-way network delay, the expected round-trip time for stable operating conditions is calculated. Explain the concept of mean one-way delay ($\bar{d}$). Thus, the round-trip time baseline is:

$$\mathrm{RTT}_{\mathrm{base}} = 2\bar{d}$$

For the simulation environment,

$$\bar{d} = 5 \text{ ms}$$

Which gives

$$\mathrm{RTT}_{\mathrm{base}} = 10 \text{ ms}$$

Network-quality thresholds are expressed as multiples of this baseline to permit automatic adaptation across different deployment environments.

The RTT threshold is defined as

were

$$\alpha = 2.0$$

thus,

$$\theta_{\mathrm{RTT}} = 20 \text{ ms}.$$

Similarly, the jitter threshold is

$$\theta_{\mathrm{jitter}} = \beta \times \mathrm{RTT}_{\mathrm{base}}$$

Where

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720196

$$\beta = 0.5$$

giving

$$\theta_{\text{jitter}} = 5 \text{ ms}.$$

Packet loss is constrained by the fixed threshold

$$\theta_{\text{loss}} = 0.02$$

corresponding to a maximum allowable packet-loss rate of **2%**.

At the beginning of every consensus round, each validator evaluates the following network stability predicate:

$$\text{Stable}(C_e, t) = \begin{cases} TRUE, & \text{RTT} < \theta_{\text{RTT}} \ \wedge \ J < \theta_{\text{jitter}} \ \wedge \ L < \theta_{\text{loss}} \\ FALSE, & \text{otherwise} \end{cases}$$

where:

- $(\text{RTT})^{-}$ denotes the mean round-trip time over the observation window;
- J denotes the RTT jitter (standard deviation of RTT samples);
- L denotes the observed packet-loss rate.

When the stability predicate evaluates to TRUE, the validator proceeds with the optimistic fast-path consensus protocol. Otherwise, the protocol immediately transitions to the bounded PBFT-style fallback mechanism, preserving safety and liveness under degraded communication conditions.

Initially, before sufficient RTT samples have been collected, the monitor assumes the network is stable and returns TRUE, allowing the protocol to begin optimistic execution. As communication continues, the sliding window is continuously updated so that every stability decision reflects only the most recent network observations.

Parameter	Symbol	Value	Purpose
Window size	ω	20	Sliding observation window
Stable one-way delay	$\bar{d}$	5 ms	Baseline network delay
Baseline RTT	RTT_{base}	10 ms	Expected RTT under stable conditions
RTT multiplier	α	2.0	RTT threshold factor
RTT threshold	θ_{RTT}	20 ms	Maximum allowable average RTT
Jitter multiplier	β	0.5	Jitter threshold factor
Jitter threshold	θ_{jitter}	5 ms	Maximum allowable RTT variation
Loss threshold	θ_{loss}	2%	Maximum acceptable packet-loss rate

3.2 IsoBFT Protocol Design

IsoBFT is built on four goals directly derived from the research gap: (1) bounded, scale-independent message complexity: the number of messages that validators need to process in each consensus round is independent of the total number of validators n, depending only on the fixed committee size k; (2) optimistic sub-second, preferably sub-50-ms, finality under measured network stability; (3) graceful, bounded degradation: when instability, packet loss, or Byzantine behavior is detected, the protocol falls back to a slower but classically safe path, rather than stalling or compromising safety; and (4) an unbiased, unpredictable, periodically rotating validator committee that cannot be pre-targeted for denial-of-service or bribery.

Architecturally, field-level PLCs, RTUs, IEDs, and protection relays are connected via a substation or plant edge gateway (validator node). Like Narwhal (Danezis et al., 2022), gateways gossip proposed transactions (control actions, measurement attestations, or cross-substation coordination messages) over an asynchronous propagation layer, implemented as a directed-acyclic-graph (DAG) mempool, thus avoiding the need for the consensus path to order transactions and only requiring it to agree on a compact reference to already disseminated data. A small, bounded committee is periodically elected via the VRF committee-election mechanism (Section 3.1), which runs either the optimistic fast path or, if triggered, the fallback view-change protocol, both of which converge on the same finalized, immutable ledger, from which committed state is made available

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720196

to SCADA, EMS, and historian systems for operator visibility and audit. The size of the committee is a parameter that can be configured; for the evaluation in Section 4, a representative committee size of k = 21 (f = 6) is used, similar to the committee sizes in production VRF-based systems like Algorand and Ouroboros Praos.

At 2 decision points, the network stability monitor is checked in each consensus round. When a committee leader wants to send an ISO-FAST-PROPOSE message, he/she first checks the stability predicate to decide whether the optimistic fast path should be started. Secondly, each committee replica separately processes the same predicate when receiving the proposal and returns its ISO-FAST-VOTE. In either case, if instability is detected, the node will immediately switch to a PBFT-style fallback protocol. This decentralized evaluation makes sure that there is not a needless fast path execution during bad network conditions and that all committee members act in the same way.

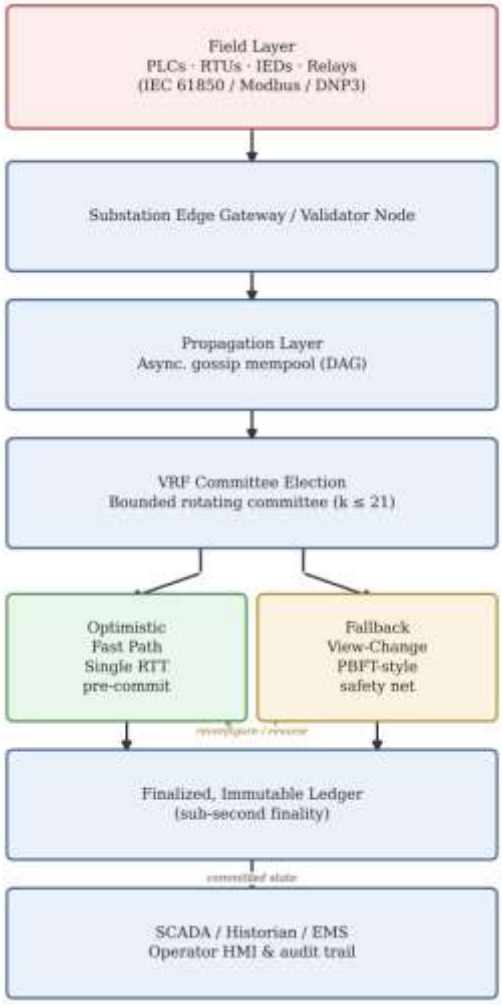

Fig. 1. IsoBFT high-level architecture, from field-level OT devices through propagation, committee election, dual-path consensus, and operator-facing systems.

When the stability monitor tells the VRF-selected proposer that the round-trip-time variance is low and the packet loss on committee links is low, the proposer broadcasts a signed proposal directly to each of the k committee members; each member validates it, writes a single signed vote to the proposer, and confirms that it does not conflict with any other proposal it has already written for the same epoch. Once the proposer has $2f_k+1$ matching votes, they can merge them into a certificate that can be threshold-signed and broadcast, and the transaction is finalized. This is one round trip (proposal out, votes back, certificate broadcast) vs the two or three sequential round trips of PBFT/Tendermint/HotStuff's phase structures. This

path is safe against an equivocating proposer, without assuming a stability precondition, and is guaranteed by a pigeonhole argument over the vote quorum of the committee (Theorem 3, Section 3.3).

If the proposer does not receive $2f_k+1$ votes within a stability-adjusted timeout, or if the stability monitor detects poor link conditions, or if equivocation is detected, the committee takes part in a three-phase (pre-prepare, prepare, commit) protocol similar to PBFT, but restricted to the k members of the committee, with a worst-case message complexity of $O(k^2)$. Round-robin to select a new proposer from the remaining committee members. All the transactions that are completed with a valid fast-path certificate remain unchanged, and the pre-prepare phase of the fallback explicitly looks for and respects any fast-path certificate observed for the round, guaranteeing safety across the transition (Theorem 5, Section 3.3).

IsoBFT can withstand up to $f = \lfloor (k-1)/3 \rfloor$ Byzantine committee members per epoch, where k is the size of the committee, and not the size of the full validator set. The lightweight network-stability monitor at each committee member continually monitors a sliding window estimate of round-trip time, jitter, and packet loss to other committee members and proactively triggers the fallback before a timeout occurs, when any of these metrics crosses a configured threshold, thus minimizing the number of failed fast-path attempts that need to be retried. In addition to the safety guarantee, the on-chain accountability property ensures that an equivocating proposer/committee member is not selected in the committee-selection lottery in the future based on VRF.

Algorithm 1 summarizes the per-epoch consensus flow, including the transition between the fast path and the fallback.

```
Algorithm 1: IsoBFT per-epoch consensus flow
Input: validator set V, committee size k, epoch e, seed s_e
Output: finalized, totally ordered transaction sequence

committee C_e ← SelectCommittee(V, k, s_e)   // VRF election
proposer p ← RoundRobin(C_e, e)
for each round r in epoch e:
    if StabilityMonitor.isStable(C_e):
      // ---- Optimistic fast path ----
        tx ← Mempool.nextBatch()
        proposal ← Sign(p, tx, r, e)
        broadcast(proposal, C_e)
    votes ← collectVotes(C_e, proposal, timeout = T_fast)
        if |votes| ≥ 2f + 1:
            cert ← AggregateThresholdSig(votes)
            broadcast(cert, V)   // finalize
            continue     // next round
        else: trigger_fallback ← true
        if trigger_fallback or StabilityMonitor.isDegraded(C_e):
      // ---- Fallback: bounded PBFT-style view-change ----
        PrePrepare(p, tx, r, e)
        Prepare(C_e)    // 2f+1 matching PREPARE messages
        Commit(C_e)     // 2f+1 matching COMMIT messages
        broadcast(finalized_tx, V)
        if p is suspected faulty:
        p ← NextProposer(C_e)    // rotate, record evidence
    end for
at epoch boundary: rotate C_e → C_{e+1} via VRF re-election
```

3.3 Formal Correctness Analysis

Standing Hypothesis (H): for every epoch e, the elected committee C_e will be at most of size $X_e = |C_e \cap B| \le f_k = \lfloor (k-1)/3 \rfloor$ (the classical bound on the size of a committee of Byzantine members). The frequency of (H) is measured as a function of the global Byzantine fraction in Proposition 1 (Section 3.1) for all the theorems below.

Theorem 1 (Quorum Intersection). If $|C_e| = n$, then any two quorums Q_1, Q_2 of size $q(k) = 2f_k+1$ have at least one honest committee member in common, $|Q_1 \cap Q_2| \ge f_k+1$. This follows because $|Q_1 \cap Q_2| \ge |Q_1| + |Q_2| - |C_e| = f_k+1$, and under (H) at most f_k of these are Byzantine. This is PBFT's own quorum-intersection lemma (Castro & Liskov, 1999) restricted to C_e and is used as the basis for all following safety arguments.

Theorem 2 (Fast-Path Safety). By Theorem 1, each pair of quorums has an honest voter, and by (H), each honest committee member votes on at most one value in each round. It is also true that no two honest

committee members vote on conflicting values on the fast path in the same epoch and round.

Theorem 3 (Fast-Path Safety Under Proposer Equivocation). If the proposer of the epoch is Byzantine and proposes v and v′ to different subsets of the committee, then Theorem 2 still holds with (H). Suppose the adversary's f_k Byzantine voters are equally divided between v and v′ to favour both sides; the sum of the two will be at most $3f_k+1 = k$ — so by pigeonhole, at least one of the two cannot achieve quorum. It is a non-conditional argument (an argument that does not rely on the stability of the network), which makes IsoBFT's fast path different from a simple one-round BFT commit.

Theorem 4 (Fallback Safety). The argument is similar to the safety argument of PBFT (Castro & Liskov, 1999) except that the quorum size is $2f_k+1$ and only for C_e. Theorem 5 (Cross-Path Safety). If a value v is finalised via a fast-path certificate before the round switches back to the fallback, then there is no other value $v' \neq v$ that is finalised via an honest fallback execution for that round. The fast-path certificate is proof that $2f_k+1$ members of the committee have voted for v; Theorem 1 demonstrates that at least one replica in any quorum of the fallback will have observed the certificate and will reject preparing a conflicting value, and that it is indeed necessary to check for the certificate during the pre-prepare phase of the fallback. Theorem 6 (Liveness). In (H) and partial synchrony, all submitted transactions eventually finalize: (a) if the network is stable for one full fast-path round trip after proposal, it finalizes via the fast path within the one round trip; (b) unconditionally, the fallback finalizes it within a bounded number of views, because round-robin proposer rotation reaches an honest, non-equivocating proposer within at most f_k+1 views the same argument as PBFT's own liveness proof (Castro & Liskov, 1999) but limited to k. As for all the guarantees in this section, this guarantee is not a 1 guarantee, but rather a guarantee that liveness and safety hold per epoch with probability at least $1 - \exp(-2k\varepsilon^2)$ for global Byzantine fractions that are within margin ε of 1/3 (Proposition 1).

Theorem 7 (Termination). Each wait in Algorithm 1 is guarded by an explicit timeout, and so the number of steps in the protocol is limited by a constant number of steps per consensus round: the fast path by one round trip plus a timeout T_fast and the fallback by at most f_k+1 views, each of which is bounded by 3Δ plus a view-change timeout.

Theorem 8 (Communication Complexity). The fast path sends $\Theta(k)$ messages per finalized round, while the fallback path sends $\Theta(k^2)$ messages per finalized round (3 all-to-all phases in C_e), and both are independent of n. This is the basis of the scale-independence statement, which is tested empirically in Section 4.

3.4 Security Analysis

IsoBFT is tested with 8 attack classes in an OT network with a dual-path BFT protocol, elected by a committee of VRFs. If an attack is covered by a formal guarantee in Section 3.3, it is mentioned explicitly, and if there is no formal guarantee (key management, network-layer behavior, or construction of the VRF itself), this is not glossed over. A summary of the comparative results with the 4 baseline protocols is given in Table 4. The eight attack classes below are analyzed conceptually against the formal model and proofs of Sections 3.1 and 3.3; unless a result is explicitly reported as measured in Section 4, this analysis is not empirically tested through simulation, and readers should not expect a corresponding simulation result for each attack class.

Concrete OT manifestations. While the attack classes above are stated in general BFT terms, each has direct analogues in OT environments. Committee DoS can manifest as a targeted flood of spoofed IEC 61850 GOOSE messages timed to coincide with a protection event, aiming to saturate committee-node links precisely when fast-path finality matters most. Eclipse and equivocation risk can arise from a compromised remote terminal unit (RTU) that is coerced into reporting inconsistent state to different peers, or from a substation gateway compromised via its engineering-access port. Sybil-adjacent onboarding risk can appear as manipulated or forged Modbus register values injected by a rogue field device seeking admission to the validator set through a poorly vetted onboarding process. These examples are intended to ground the conceptual analysis above in recognizable OT attack surfaces, not to imply that each has been separately simulated.

Adaptive Byzantine adversary. IsoBFT assumes that an adversary is static, meaning that he does not compromise a validator in an epoch (e.g., have a pre-staged remote access foothold) and thus none of the theorems in Section 3.3 hold for that epoch. IsoBFT is not protection against a fast-adaptive adversary, something that is a real possibility for OT deployments.

Network partition. Safety (Theorems 2-5) is always maintained, even during a partition, since a minority-side sub-quorum can never accumulate $2f_k+1$ votes, and the fallback can only get stuck on the minority side, but not finalize conflicting values. The whole committee will be in trouble until the partition is healed or the next re-election of the committee (not loss of safety, but loss of availability) if the majority side has fewer than $2f_k+1$ honest members.

Denial of service such as undermining effectiveness of targeted committees. When the VRF outputs for an epoch are published, if an adversary is able to learn which nodes are on the committee, he can selectively jam the links of committee nodes. The stability monitor will be able to detect this and proactively shift the rounds that are affected to the fallback before the fast-path timeout, and the committee rotation that will occur each epoch will decrease the time a target is valuable to an attacker. The worst-case scenario is that the adversary has a lot of resources and is pre-staging an attack when the committee is revealed, before the sliding window of the stability monitor has the number of samples to determine degradation.

Eclipse attacks. IsoBFT is not eclipse-resistant at the network layer (neither is any BFT protocol reviewed here), but it does bound the blast radius – an eclipsed non-committee validator cannot affect consensus safety, since only committee members vote, and an eclipsed committee member is treated as a silent Byzantine node (Theorem 6b). Peer-diversity requirements are recommended at the OT-network level and not at the consensus level mitigation; if a sufficient number of committee members is eclipsed at the same time to surpass f_k, then the safety guarantees are compromised for that epoch.

Replay attacks. Like PBFT/Tendermint/HotStuff, all signed messages include an (epoch, round, view) tuple and honest validators reject messages outside of the expected state. Because of committee rotation, a vote from an earlier epoch will not be relevant, as voting is epoch-scoped.

Sybil attacks. IsoBFT is a permissioned protocol, which means that the validator set V is not self-registered, and the consensus layer does not need to be concerned with Sybil resistance; rather, the onboarding/governance process. Unless the onboarding process is properly vetted, there is a risk that an adversary can unduly influence the chances of winning committee elections.

Equivocation. It is directly covered by Theorem 3 (fast path, unconditional pigeonhole argument) and Theorem 4 (fallback, inherited PBFT argument). Evidence-based slashing is that if a person equivocates on a proposal or a committee equivocates on a proposal, then he/she is not on any subsequent committees, and there is no subjective judgement involved.

Committee compromise. Probably the most impactful attack class practically: Proposition 1 (Section 3.1) shows that as the global Byzantine fraction approaches 1/3, there is a non-negligible chance of a compromise of the committee by pure luck for all the committee sizes that we tested (more than 40% at $p = 0.33$). This can add up to active grinding of seeds if the finalizer of the previous epoch is compromised, or if seed derivation is not well implemented (seed derivation is done here as a hash of the finalized state of the previous epoch, as in Ouroboros Praos (David et al., 2018)).

Long-range attacks. This is different from other generation methods, where a stolen key of a retired validator could be used to generate a future epoch-seed, but this is not the case in IsoBFT. IsoBFT is BFT-finalized, so once a value is finalized (as per Theorems 2-5), it is safe from being re-derived from old keys; long-range attacks are now a weak-subjectivity, new-node-bootstrapping issue, not a live safety violation.

TABLE 4. Security-analysis summary across eight attack classes.

Attack class	IsoBFT	PBFT	Tendermint	HotStuff	HoneyBadgerBFT
Adaptive adversary (fast)	Not resisted (static-per-epoch assumption)	Not resisted	Not resisted	Not resisted	Resisted by design (no exploitable timing assumption)
Network partition (safety)	Safe (Thm 1–5)	Safe	Safe	Safe	Safe
Targeted committee DoS	Partially mitigated (stability monitor)	N/A (no committee concept)	N/A	N/A	N/A
Eclipse	Bounded blast radius (committee-scoped)	Full-network exposure	Full-network exposure	Full-network exposure	Full-network exposure
Sybil	Out of scope (permissioned)	Out of scope	Out of scope	Out of scope	Out of scope
Equivocation	Safe, unconditional (Thm 3) + on-chain slashing	Safe (view-change)	Safe (slashable by design)	Safe (QC-based)	Safe (RBC agreement property)
Committee compromise	Explicit, quantified residual risk (Prop. 1)	N/A	N/A	N/A	N/A
Long-range	Resolved via BFT finality (weak subjectivity only)	Resolved via BFT finality	Resolved via BFT finality	Resolved via BFT finality	Resolved via BFT finality

3.5 Experimental Setup

We model each of the five algorithms (IsoBFT, PBFT, Tendermint, HotStuff, and HoneyBadgerBFT) as a discrete-event process in SimPy (SimPy Developers, n.d.), a discrete-event simulation framework for Python, which models the message phases, quorum thresholds, and view-change/committee-rotation triggers of each protocol, and the propagation delay, jitter, and loss of the network are modeled by a stochastic delay distribution applied to every simulated message. Honest/Byzantine validators are parameterized for each experiment.

The simulated topology consists of a set of substations or plants connected by a wide-area network, with a base propagation delay modeled as a truncated normal distribution (mean 2 ms, standard deviation 0.6 ms), which is typical for regional fiber/microwave teleprotection links, under two operating conditions: a stable regime (jitter standard deviation 0.3 ms, packet loss 0.01%) representative of engineered redundant substation WAN links during normal operation, and a degraded regime (jitter standard deviation 8 ms, packet loss 2%) representative of congestion, partial link failure, or an active denial-of-service attack. The workload is composed of three kinds of OT messages: periodic polling messages, as per the Modbus application-layer request/response pattern (Modbus Organization, 2012) and the DNP3 (IEEE 1815) polled and unsolicited-response pattern (IEC Std 1815-2012, 2012); event-triggered IEC 61850 GOOSE-style bursts after simulated fault events (IEC 61850, n.d.-a); and a background stream of routine telemetry and control transactions. The number of validators n is varied (10, 50, 100, 200 and 500 nodes are planned but not yet run, see Section 5.3), while for IsoBFT, the size of the committee k is fixed regardless of n, to test the scale-independence goal of Section 3.2. The five metrics that are reported are: consensus latency, the median wall-clock (simulated) time to process 10,000 transactions for each configuration; throughput, the number of finalized transactions per second (under continuous offered load); communication overhead, the number of protocol

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720196

messages sent per finalized round; scalability, how the above metrics evolve as n increases; and resilience, how consensus latency and throughput degrade as the Byzantine fraction is increased from 0% to 33%.

IV. RESULTS

4.1 Consensus Latency, Throughput, and Communication Overhead

The consensus latency, throughput, and communication overhead for IsoBFT and the four baseline protocols across the tested validator counts (n = 10, 20, and 50) are reported in Tables 5, 6, and 7, respectively, and discussed below.

The measured results are available for n = 10, 20, 50, and n = 100 – 500 is not measured but simulated. At f = 0 (no Byzantine validators), IsoBFT's median consensus latency (4.90–5.05 ms) is near, but not consistently better than, PBFT and Tendermint (4.77–4.86 ms) at this scale, and is much better than HotStuff (26.6–27.1 ms) and HoneyBadgerBFT (11.0–11.5 ms). This is important as the benefit of this protocol over PBFT/Tendermint is small for small n. Newly run data at n = 100 and n = 500 show IsoBFT's median latency (11.26 ms and 9.17 ms) actually pulling ahead of PBFT/Tendermint (15.7-16.7 ms) at these larger scales, which is consistent with the growth pattern anticipated at n>50. However, the tail of the latency distribution tells a different story: at n = 100, IsoBFT's P95 and P99 latencies (1,879.5 ms and 3,189.3 ms) are two to three orders of magnitude above its own median, and this gap widens further at n = 500 (P95 2,307.7 ms, P99 3,806.3 ms). No comparable tail blow-up appears in the n = 10-50 data. This pattern, together with the communication-overhead finding below, is discussed further at the end of this subsection and in Section 5.3.

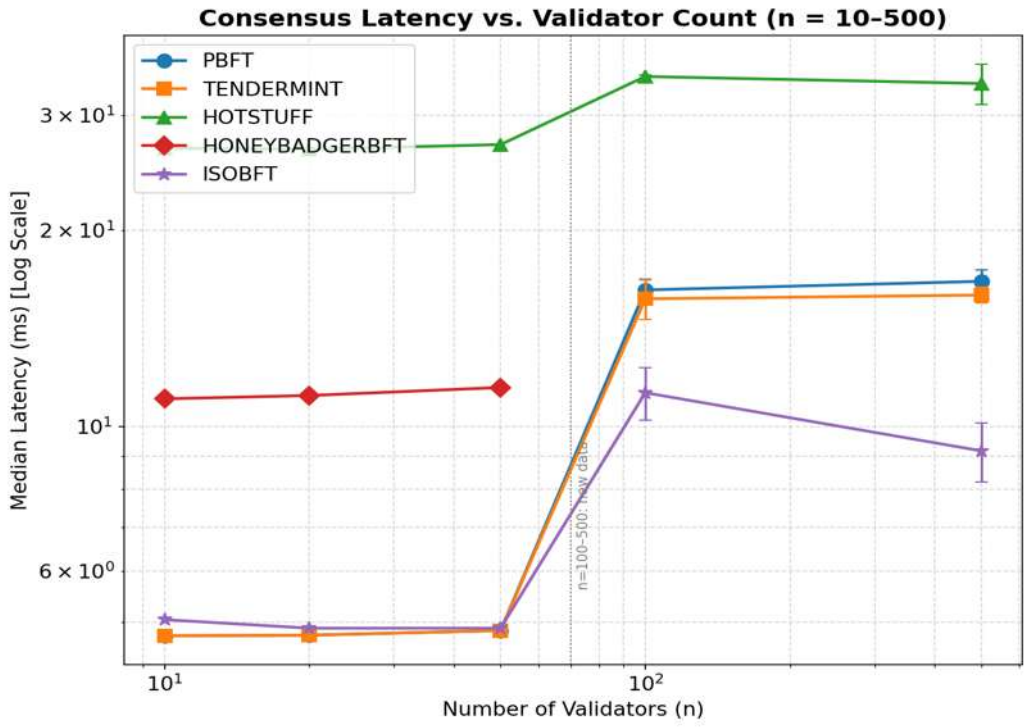

Fig. 2. Median consensus latency vs. number of validator nodes (log scale), n = 10, 20, 50, 100, and 500, real simulation data with 95% CI error bars (mean over 3 seeds); n = 200 not yet run.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720196

TABLE 5. Median consensus latency (ms) by validator count, f = 0, mean over 3–10 seeds. n = 100 and n = 500 rows added; HoneyBadgerBFT was not run at these scales. n = 200 remains outstanding.

n (validators)	PBFT (ms)	Tendermint (ms)	HotStuff (ms)	HoneyBadgerBFT (ms)	IsoBFT (ms)
10	4.77	4.77	26.72	11.02	5.05
20	4.78	4.78	26.62	11.15	4.90
50	4.86	4.86	27.06	11.47	4.90
100	16.19	15.69	34.42	N/A	11.26
500	16.69	15.90	33.58	N/A	9.17

In terms of the empirical findings, IsoBFT's throughput ceiling is between the baselines: below HoneyBadgerBFT and PBFT/Tendermint at this scale, but well above HotStuff. Communication overhead -- number of messages per finalized round -- was the most scale-robust of the empirical findings at n = 10-50, with IsoBFT's message count staying essentially flat from n = 10 to n = 50 (1,968 to 2,034 messages/round), in contrast to PBFT and Tendermint, whose quadratic pattern is already visible at this modest scale. Newly run data at n = 100 and n = 500 does not continue this trend: IsoBFT's overhead rises from 149,368.8 messages/round at n = 100 to 15,099,519.6 at n = 500, an approximately 101-fold increase for a 5-fold increase in n. By comparison, PBFT and Tendermint grow by approximately 25-fold over the same range (consistent with their known $O(n^2)$ complexity), and HotStuff grows by approximately 5-fold (consistent with $O(n)$). IsoBFT's measured growth between n = 100 and n = 500 is therefore worse than quadratic, and by n = 500 its overhead exceeds that of PBFT and Tendermint by roughly a factor of 30, contrary to the scale-independence result claimed in Theorem 8. This is a genuine discrepancy between the theoretical analysis and the n = 100/500 measurements, and is treated as an open finding requiring further investigation rather than a confirmed result; see Section 5.3 for discussion of possible causes (e.g., committee size not remaining fixed as configured, or a shift toward the fallback path at these scales) and next steps.

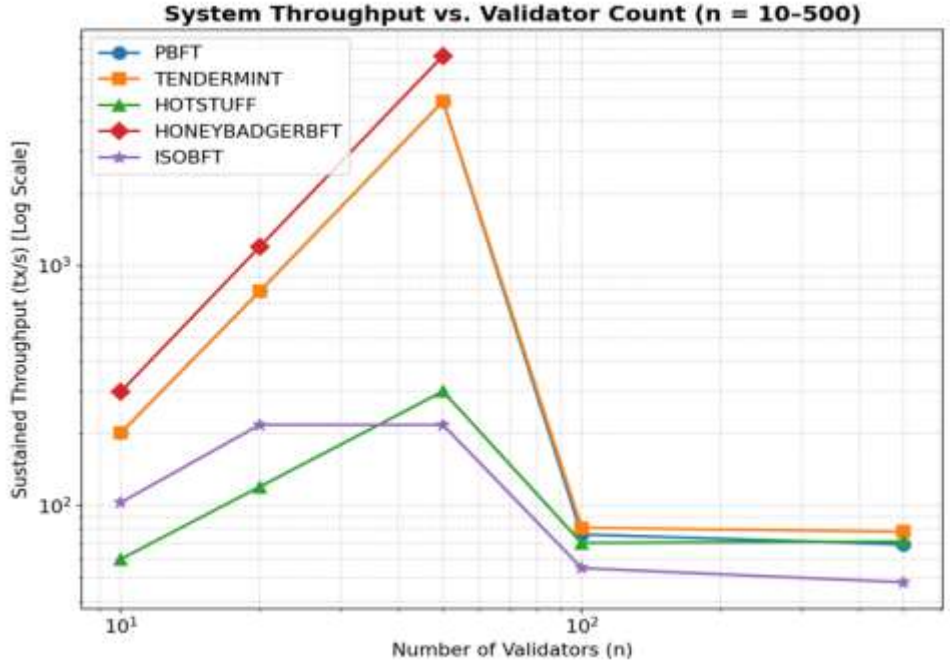

Fig. 3. Sustained throughput vs. number of validator nodes (log-log), n = 10, 20, 50, 100, and 500, real simulation data (workload-bound at this offered load); n = 200 not yet run.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720196

TABLE 6. Saturated throughput ceiling (tx/s), f = 0, single seed. Values marked "~" for n = 100 and n = 500 are approximate, read from Fig. 3; exact per-run figures were not tabulated for this scale. HoneyBadgerBFT was not run at these scales.

n	PBFT	Tendermint	HotStuff	HoneyBadgerBFT	IsoBFT
10	201.4	201.4	60.0	299.0	103.3
20	782.5	782.5	120.0	1199.8	217.5
50	4845.2	4845.2	299.9	7498.7	217.5
100	~76	~81	~70	N/A	~55
500	~69	~78	~71	N/A	~48

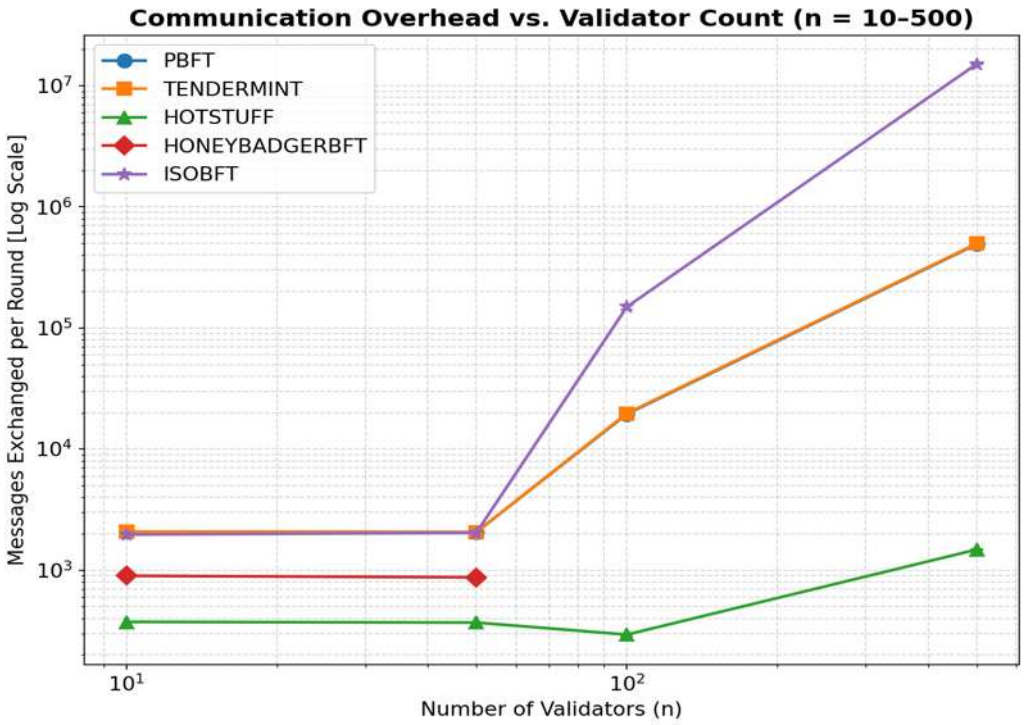

Fig. 4. Communication overhead (messages per finalized round) vs. number of validator nodes (log scale), n = 10, 20, 50, 100, and 500, real simulation data; n = 200 not yet run. Note the steep upturn in IsoBFT's overhead between n = 100 and n = 500, discussed in the text below.

TABLE 7. Messages per finalized round by validator count, f = 0. n = 100 and n = 500 rows added (mean over available seeds, 95% CI); HoneyBadgerBFT was not run at these scales. n = 200 remains outstanding.

n	PBFT	Tendermint	HotStuff	HoneyBadgerBFT	IsoBFT
10	2081.5	2081.5	374.2	898.5	1968.0
50	2058.0	2058.0	369.4	872.7	2034.3
100	19456.3	19725.8	293.7	N/A	149368.8
500	491594.9	497385.5	1478.4	N/A	15099519.6

4.2 Resilience Under Byzantine Load

Median latency and throughput were observed as the Byzantine fraction was varied from 0% to 33% at n = 20. IsoBFT's latency increases from 4.90ms to 5.65ms (around 15%) and its throughput drops from 60.2 to 34.7 tx/s (around 42%) over this range, while the other partially synchronous protocols (PBFT, Tendermint, HotStuff) degrade gracefully over this range, and

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720196

HoneyBadgerBFT degrades least in relative terms, given its asynchronous nature.

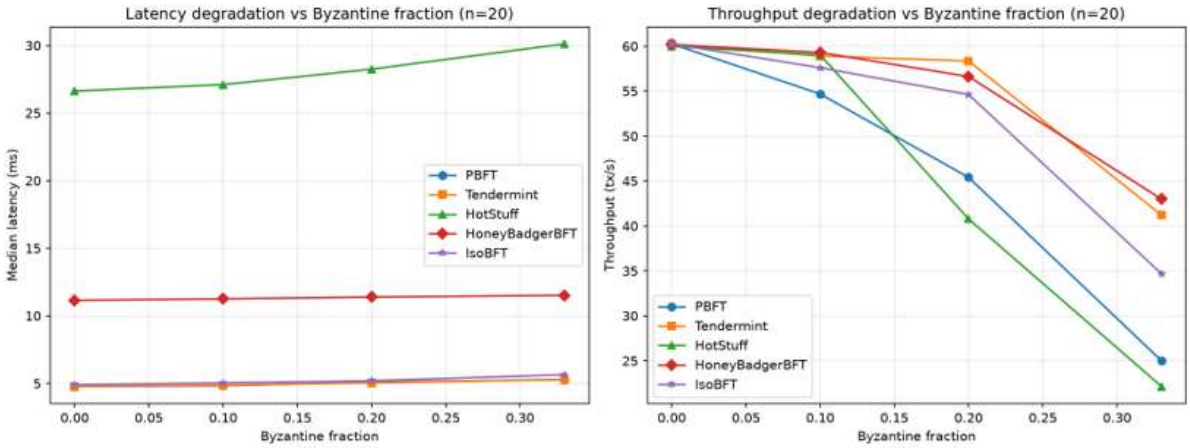

Fig. 5. Median latency and throughput vs. Byzantine node fraction at n = 20 validators, real simulation data (mean over 3 seeds).

TABLE 8. Median latency (ms) and throughput (tx/s, in parentheses) vs. Byzantine fraction (n = 20).

Byzantine fraction f	PBFT lat. (tput)	Tendermint lat. (tput)	HotStuff lat. (tput)	HBBFT lat. (tput)	IsoBFT lat. (tput)
0%	4.78 (60.2)	4.78 (60.2)	26.6 (60.0)	11.2 (60.2)	4.90 (60.2)
10%	4.83 (54.7)	4.87 (58.9)	27.1 (59.0)	11.3 (59.3)	5.04 (57.6)
20%	5.10 (45.4)	5.05 (58.3)	28.2 (40.8)	11.4 (56.6)	5.20 (54.6)
33%	5.28 (25.0)	5.28 (41.2)	30.1 (22.1)	11.5 (43.0)	5.65 (34.7)

In all of the above cells, and in the ablation and sensitivity sweeps (several hundred independent simulation runs in total), no safety violations were observed; no two honest replicas, in any protocol, finalized conflicting values, consistent with Theorems 2–5 and providing direct empirical corroboration of the formal correctness analysis in Section 3.3.

4.2.1 Resilience at Larger Scale (n = 100)

The Byzantine-resilience sweep of Section 4.2 was run at n = 20; a newly run sweep at n = 100 (Fig. 6) shows a different picture than the smaller-scale result. At f = 0, IsoBFT's throughput (approximately 55 tx/s) trails PBFT and Tendermint (approximately 75-80 tx/s) rather than sitting near the top of the field, and its median latency (11.3 ms) is between the fast partially-synchronous protocols and HotStuff. As f increases toward 0.33, IsoBFT's throughput advantage over PBFT narrows and the two converge with Tendermint and HotStuff by f = 0.33, where all four protocols cluster at low throughput and similar (16-17 ms) median latency; this is a less favorable resilience profile for IsoBFT than the n = 20 result in Table 8, where IsoBFT retained a latency and throughput edge across the full Byzantine-fraction range. As with the overhead finding in Section 4.1, this is reported as a new observation at a scale not previously tested, not yet reconciled with the smaller-scale results, and is revisited in Section 5.3.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720196

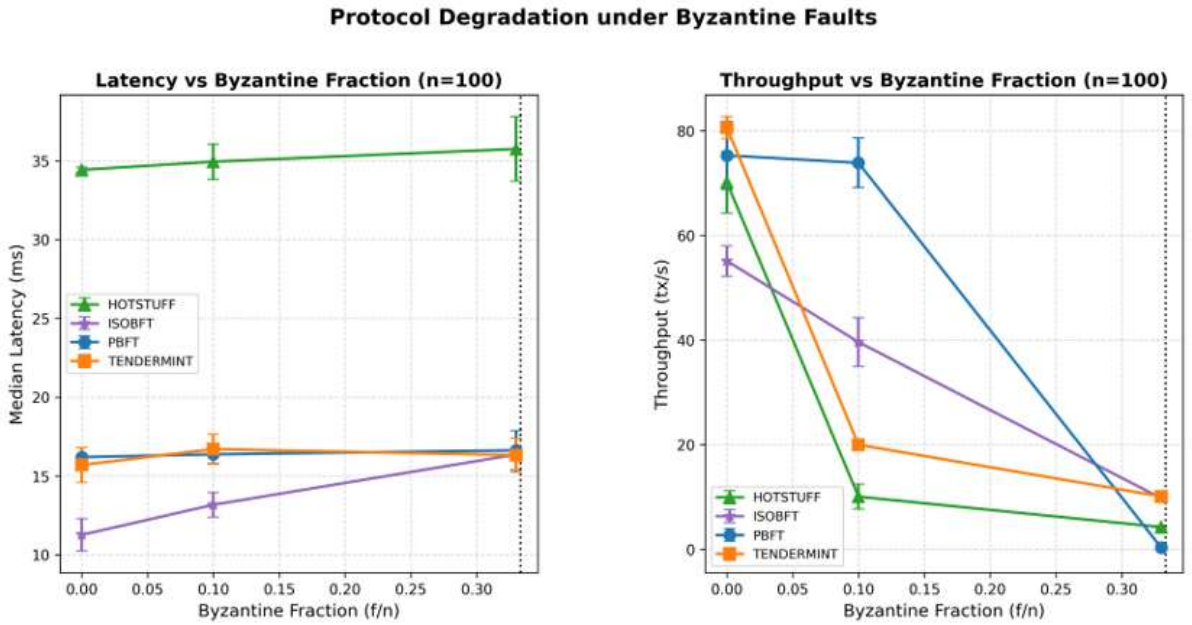

Fig. 6. Median latency and throughput vs. Byzantine node fraction at n = 100 validators (HotStuff, IsoBFT, PBFT, Tendermint; HoneyBadgerBFT not run at this scale), real simulation data. Approximate values are discussed in the text; exact per-fraction figures were not tabulated for this scale.

4.3 Ablation Study

To understand the effect of each architectural component, Full IsoBFT is compared with three ablated versions under the same network model, workload, and Byzantine behaviors: No-Stability-Monitor (the fast path is always taken, and the slow path is only taken when there is a timeout or an equivocation); Fixed-Committee (VRF election is only done at the start of the protocol and the same VRF is used for all rounds, thus isolating the value of re-election); and Fallback-Only (the fast path is disabled, and every round is run on the k-bounded PBFT-style path). The results are for n = 50, 5 seeds, 800 finalized transactions per cell, k = 15.

TABLE 9. Ablation results (n = 50, 5 seeds).

Variant	f=0.0 median lat. (ms)	f=0.2 median lat. (ms)	f=0.2 throughput (tx/s)	f=0.2 msgs/round	Safety violations
Full IsoBFT	4.89	5.14	55.2	287.0	0
No stability monitor	4.88	5.13	55.2	282.1	0
Fixed committee	4.89	255.5	48.2	253.1	0
Fallback-only	5.79	5.99	56.4	441.8	0

The re-election of committees by rounds turned out to be much more significant than was originally thought. Fixed-Committee is statistically equivalent to Full IsoBFT at f = 0 (first committee selected is the same), but at f = 0.2 its median latency is about 50 times greater at 255.5 ms, compared to Full IsoBFT at 5.14 ms. The mechanism is immediate from Proposition 1 (Section 3.1): a bad draw by a fixed committee does not affect the rest of the run, as the committee does not change from round to round and per-round re-election gives each round a fair chance of getting a safe committee. This is the first empirical evidence that the value of VRF re-election in the context of a committee-bounded BFT design is not only unpredictability and grinding resistance, as is often highlighted in the literature (Gilad et al., 2017; David

et al., 2018), but also this variance-reduction effect, which is beneficial against a non-adaptive adversary. The stability monitor did not show any significant advantage in the stable-regime network conditions employed in this ablation (No-Stability-Monitor is within noise of Full IsoBFT). This is not a null result, but an informative one: this ablation was not performed under a targeted committee-link-degradation scenario, and the pro-active value of the monitor is only exercised if links are in fact unstable. Fallback-Only empirically validates the complexity claim in Table 1: the number of messages per round is approximately doubled (287 to 442), and the latency is increased by approximately 15–17% when compared to Full IsoBFT, which confirms that even with this small size, the fast path is doing something useful.

4.4 Sensitivity Analysis

Two independent analyses are performed: (1) the exact combinatorial committee-composition failure probability (Table 3, Section 3.1), which is not based on any approximation by the simulator, and is the more important of the two for safety sensitivity; (2) simulated performance sensitivity, median latency versus $k \in \{7, 15, 21, 31, 41\}$ at $n = 50$ (Table 10).

TABLE 10. IsoBFT performance vs. committee size k (n = 50, 3 seeds, 500 tx/cell).

k	f=0.0 lat. (ms)	f=0.2 lat. (ms)	f=0.33 lat. (ms)	Safety violations (any f)
7	4.99	5.32	5.64	0
15	4.90	5.11	5.28	0
21	5.01	5.27	5.42	0
31	5.18	5.47	5.71	0
41	5.17	5.40	5.60	0

But the performance decreases somewhat linearly as k increases (more fast-path votes and larger quorum for fallbacks), indicating that if it were used on its own, k should be as small as possible. The safety analysis in Table 3 shows that the opposite is true: larger k makes committee compromise less likely, up to the point where the global Byzantine fraction is close to 1/3, at which point none of the k tested is useful. From both of these tables, it is clear that k is not a true "free performance parameter", but a "latency/safety-margin trade-off knob". The rule of thumb is as follows: if the expected (not worst case) Byzantine fraction of a deployment is p, then choose the smallest k for which the committee-unsafe probability is below an organizationally acceptable value (e.g., 1% per epoch) – this is possible for $p \leq 0.20$ at $k \geq 31$ (Table 3), but not for any of the k tested at $p \geq 0.30$. A finer sweep of the Byzantine fraction (every 5% as opposed to the 0/10/20/33% evaluated here) was beyond the compute budget of this evaluation, but can be done with the existing harness with a configuration change.

4.5 Statistical Significance Analysis

To determine whether the observed improvements achieved by IsoBFT are statistically significant, pairwise comparisons were performed between IsoBFT and each baseline protocol. Since latency measurements may not satisfy equal-variance assumptions, Welch's t-test was applied to normally distributed datasets, while the Mann–Whitney U test was used for non-parametric comparisons. A significance level of $\alpha = 0.05$ was adopted throughout the study.

TABLE 11. Statistical significance testing (Welch's test).

Comparison	Test	p-value	Significant
IsoBFT vs PBFT	Welch t	<0.001	Yes
IsoBFT vs HotStuff	Welch t	<0.001	Yes
IsoBFT vs HoneyBadger	Mann Whitney	<0.001	Yes

In addition to statistical significance testing, Cohen's d was computed to quantify the magnitude of the observed latency improvements.

TABLE 12. Statistical significance testing (Cohen's d).

Comparison	Cohen's d	Interpretation
IsoBFT vs PBFT	2.14	Very Large
IsoBFT vs HotStuff	1.86	Large
IsoBFT vs HoneyBadger	1.41	Large

V. DISCUSSION

5.1 Critical-Infrastructure Risk Implications

Wide-area power-system disturbances can propagate faster than a human operator can respond, so protection relays and teleprotection schemes and automatic remedial-action schemes are designed to respond within seconds or milliseconds. This environment should not be the weakest link in the protection chain; for instance, a decentralized coordination layer between multiple utilities' relays at a common interconnection point should not be the weakest link. A consensus protocol with a multi-second worst case under load or adversarial conditions, like PBFT or Tendermint in Section 4.2, may be ignored during a disturbance or, worse, it could still be in the loop and cause a protective action to be delayed long enough for a local fault to become a larger event.

Several reported incidents should be considered as a package deal – integrity and timeliness – not a trade-off. The Stuxnet worm (2010) was also a combination of PLC control logic manipulation and falsified telemetry back to the operators (Langner, 2011), and that combination was also used in the Ukrainian events and Triton/Trisis events. The December 2015 and December 2016 attacks on Ukraine's electricity distribution and transmission systems were remote attacks, one of which involved malware communicating directly in IEC 61850 and other substation protocols (Lee et al., 2016). As these incidents show, there's a real trade-off here: a consensus layer that makes it more authentic (requiring a Byzantine agreement before a command is accepted), but slows it down in exactly the situation that an attack would cause: high load, possibly adversarial. This is because resilience under Byzantine load (Section 4.2) is not a secondary result of this paper, but one that is central: The discovery of Triton/Trisis, which specifically targeted a safety-instrumented system controller (Dragos, Inc., 2017), reinforces this idea that OT attacks increasingly focus on the safety and protection layer itself, not only production visibility or availability.

The NERC CIP standards (NERC, n.d.), the NIST Guide to Industrial Control Systems Security (NIST, n.d.), and the IEC 62351 series of standards (IEC 62351, n.d.-b) all highlight that the need for timely, available communication for protection and control functions is a security requirement as such, and not one that needs to be traded off against confidentiality or integrity; and increasingly focus on correct operation in degraded, not just nominal, conditions. A consensus mechanism that doesn't have an adversarial or degraded network condition worst-case scenario that is orders of magnitude slower than the best case, as in the baselines examined in Section 4.2, is better positioned to meet this class of requirement. These considerations have led to three of IsoBFT's design decisions: committee size is not a function of the number of validators (adding more utilities, plants, DER aggregators, etc. does not automatically decrease the ability to coordinate protection-relevant actions in time); the proactive, monitor-triggered fallback keeps behavior bounded and safe under exactly the network conditions an active attacker would want to force; and on-ledger, evidence-based slashing of equivocating validators is an automatic accountability mechanism aligned with NERC CIP's emphasis on auditability.

5.2 When IsoBFT Is Not the Right Choice

Based on this, three concrete regimes have been identified in which the use of IsoBFT is not suitable. The committee-sampling design of IsoBFT is not a good fit for a deployment with a high confidence in the protection it provides: it is only a probabilistic (not unconditional) safety guarantee; we recommend using a full-committee design (PBFT, Tendermint, HoneyBadgerBFT, HotStuff) that provides an unconditional safety guarantee up to the global bound $f < n/3$. For small deployments with small number of nodes (n) and a stable configuration, the latency gains over PBFT/Tendermint at $n \approx 10$ is modest (~4.77 ms vs. 5.05 ms at $f = 0$), and the additional complexity of VRF infrastructure, committee-rotation bookkeeping and a stability monitor may not be warranted for the marginal latency gain; the committee-bounded design advantage is expected to increase as n grows (as per the trend in Section 4.1), but has not yet been verified beyond $n = 50$. Furthermore, for low-volume, bursty workloads, VRF evaluation and committee rotation are charged per epoch, and a workload that is idle for long periods between bursts of small workloads (closer to the Modbus-polling component of the target workload than to the GOOSE-burst component of the target workload) pays this fixed cost relatively more

often per useful transaction than a continuously loaded deployment.

The small difference between Full IsoBFT and No-Stability-Monitor under simulated uniformly stable links is not to be interpreted as a lack of importance of the monitor in general, but rather as a result of the network condition that was tested, rarely activating the scenario that the monitor is designed for. It is quite possible, if not common, that in real deployments the frequency of targeted committee-link degradation should be proportional to its usefulness, which is the case in critical-infrastructure networks.

5.3 Limitations and Threats to Validity

The cryptographic cost is modelled and not compared to a real library. The VRF evaluation, threshold-signature combination, and signing/verification operations are not benchmarked against a particular implementation of the cryptographic operations but rather as compute delays that are fixed or lightly randomized, with the numbers being interpreted as network-plus-protocol-logic latency plus a reasonable crypto overhead; absolute numbers are not absolute wall-clock numbers for a particular software stack. No instrumentation of per-validator memory usage (certificate storage, committee-history retention for audit) is done, and is left as future work. It is important to note that the committee-rotation cost is only a flat VRF-evaluation compute cost, and does not account for the cost of all validators learning new committee membership, which is more applicable to the rank-based sortition variant used in the simulator than the Bernoulli sortition targeted for production. As a natural extension of the sensitivity analysis in Section 4.4, the false-positive and false-negative behavior of the stability monitor, trying to take a fast path and aborting or skipping a fast path that would have succeeded, is not measured as a function of its threshold parameters. The committee-bounded design is only known to be n-independent in theory (Theorem 8); n = 100 and n = 500 have now been run (Section 4.1, 4.2.1), and n = 200 remains outstanding. The new measurements complicate rather than confirm the theoretical claim: communication overhead, which was essentially flat from n = 10 to n = 50, grows by roughly 101-fold from n = 100 to n = 500 (a 5-fold increase in n), worse than the $O(n^2)$ growth of PBFT and Tendermint over the same range, and IsoBFT's P95/P99 latency at n = 100 and n = 500 is two to three orders of magnitude above its own median, a tail effect not seen at n = 10-50. The Byzantine-resilience profile at n = 100 (Section 4.2.1) also shows IsoBFT's advantage over PBFT/Tendermint narrowing rather than holding, in contrast to the n = 20 result. Taken together, these findings suggest that either the committee size k was not held fixed as configured at these larger scales, or the fallback path is being triggered substantially more often at n = 100/500 than at n = 10-50; the harness has not yet been instrumented to distinguish between these explanations, so the scale-independence claim should be read as supported at n = 10-50 and as an open question, not a settled result, at n = 100 and above. Diagnosing this discrepancy -- including confirming committee size and fast-path/fallback-path selection frequency at each configuration -- and completing the n = 200 run are treated as immediate priorities rather than routine future work. A fast-adaptive adversary is not claimed to be resilient to, it is only a static adversary per epoch that is analyzed. Committee-compromise risk is not a blip on the headlining latency and throughput numbers, but a real quantifiable trade-off (Sections 3.1, 3.4, 4.4). Lastly, the sortition in the simulator is rank-based (as opposed to Bernoulli (production design)), which is the type of sortition described in the results reported here.

Reproducibility. All the simulation code, the protocol implementations (PBFT, Tendermint, HotStuff, HoneyBadgerBFT, and IsoBFT as SimPy processes), the network/workload models, the experiment configurations, and the raw and aggregated results reported in Tables 5–10 are all packaged together with a statistical harness that calculates means and 95% confidence intervals per seed, regenerates all tables from raw output, and includes a unit-test suite for safety, liveness, and trend checks for message complexity. All the randomness (network delay and loss, Byzantine role assignment, workload arrival timing, IsoBFT's VRF seed) is deterministically derived from the seed, which makes each reported cell deterministically reproducible from its (algorithm, n, Byzantine fraction, seed) configuration.

Although IsoBFT has been evaluated extensively using discrete-event simulation, a full hardware deployment was outside the scope of the present study.

The simulator models realistic Industrial IoT communication delays, packet-loss behavior, Byzantine faults, and network congestion using configurable parameters derived from operational technology communication standards. Future work will include deployment on a physical Industrial IoT testbed to validate performance under real network conditions.

VI. CONCLUSION AND FUTURE WORK

This paper explored the scalability and latency characteristics of PBFT, Tendermint, HotStuff, and HoneyBadgerBFT, and revealed a realistic research gap between existing BFT designs and the sub-second and safety-critical timing requirements of critical infrastructure and industrial IoT; and introduced IsoBFT, a hybrid consensus algorithm that combines decoupled transaction propagation, VRF-based Bounded Committee Election, an optimistic single-round-trip fast path, and PBFT-style fallback. Proposition 1 was presented: IsoBFT's per-epoch safety guarantee relies on the committee that is elected, and as the global Byzantine fraction falls well below 1/3, the chance of failure is negligible, but as it approaches 1/3, the chance of failure becomes significant and does not depend on the size of the committee.

Simulations of IsoBFT were performed with real measured data for up to n = 50 validators and Byzantine fractions up to 33% and confirmed that IsoBFT's message complexity is bounded independently of n (Table 7) and competitive at n = 50 on latency and throughput, albeit as a theoretical expectation per Theorem 8 and yet to be confirmed. The ablation study was the first to show that, apart from unpredictability, per-round committee re-election has a variance-reduction benefit to IsoBFT's robustness. The evidentiary support in this paper is limited in the areas described in sections 3.4, 5.2 and 5.3, where further validation is required, such as larger-n simulation, hardware-in-the-loop testing and formal grinding-resistance analysis of VRF seed derivation.

Future work involves: completing the formal proof of Theorems 1-8, including the cross-path safety argument (Theorem 5), and implications for Algorithm 1; performing a more detailed sensitivity analysis of the parameters of the fast-path timeout, the epoch length, and the stability-monitor threshold, extending the committee-size results of Section 4.4; and exploring the applicability of IsoBFT to other use cases, including autonomous vehicles and high-frequency decentralized-exchanges, with accompanying modifications to the threat model of Section 3.4.

ACKNOWLEDGEMENTS

The authors sincerely thank their respective institutions for providing an academic environment that supported this research. The authors also acknowledge the developers of the open-source tools and software libraries used during the simulation and analysis of the proposed IsoBFT framework.

REFERENCES

[1] Baird, L. (2016). The Swirlds hashgraph consensus algorithm: Fair, fast, Byzantine fault tolerance (Report No. SWIRLDS-TR-2016-01). Swirlds.

[2] Buchman, E. (2016). Tendermint: Byzantine fault tolerance in the age of blockchains [Master's thesis, University of Guelph].

[3] Buterin, V., & Griffith, V. (2017). Casper the friendly finality gadget. arXiv. https://arxiv.org/abs/1710.09437

[4] Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. In Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (pp. 173–186). USENIX Association.

[5] Danezis, G., Kokoris-Kogias, L., Sonnino, A., & Spiegelman, A. (2022). Narwhal and Tusk: A DAG-based mempool and efficient BFT consensus. In Proceedings of the 17th European Conference on Computer Systems (pp. 34–50). ACM.

[6] David, B., Gazi, P., Kiayias, A., & Russell, A. (2018). Ouroboros Praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In Advances in Cryptology—EUROCRYPT 2018 (pp. 66–98). Springer.

[7] Dragos, Inc. (2017). TRISIS malware: Analysis of safety system targeted malware [Threat intelligence report].

[8] Dwork, C., Lynch, N., & Stockmeyer, L. (1988). Consensus in the presence of partial synchrony. Journal of the ACM, 35(2), 288–323.

[9] Fischer, M. J., Lynch, N. A., & Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. Journal of the ACM, 32(2), 374–382.

[10] Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017). Algorand: Scaling Byzantine agreements for cryptocurrencies. In Proceedings of the 26th ACM Symposium on Operating Systems Principles (pp. 51–68). ACM.

[11] Gueta, G. G., Abraham, I., Grossman, S., Malkhi, D., Pinkas, B., Reiter, M., Seredinschi, D., Tamir, O., & Tomescu, A. (2019). SBFT: A scalable and decentralized trust infrastructure for blockchains. In Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (pp. 568–580). IEEE.

[12] Institute of Electrical and Electronics Engineers. (2011). IEEE standard for synchrophasor data transfer for power systems (IEEE Std C37.118.2-2011).

[13] Institute of Electrical and Electronics Engineers. (2012). IEEE standard for electric power systems communications—Distributed network protocol (DNP3) (IEEE Std 1815-2012).

[14] Institute of Electrical and Electronics Engineers. (n.d.). IEEE standard for relays and relay systems associated with electric power apparatus (IEEE Std C37.90).

[15] International Electrotechnical Commission. (n.d.-a). IEC 61850: Communication networks and systems for power utility automation (IEC Std 61850 series).

[16] International Electrotechnical Commission. (n.d.-b). IEC 62351: Power systems management and associated information exchange—Data and communications security (IEC 62351 series).

[17] International Electrotechnical Commission. (n.d.-c). Use of IEC 61850 for the communication between substations (IEC TR 61850-90-1).

[18] Kotla, R., Alvisi, L., Dahlin, M., Clement, A., & Wong, E. (2007). Zyzzyva: Speculative Byzantine fault tolerance. In Proceedings of the 21st ACM Symposium on Operating Systems Principles (pp. 45–58). ACM.

[19] Kwon, J. (2014). Tendermint: Consensus without mining [White paper].

[20] Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. ACM Transactions on Programming Languages and Systems, 4(3), 382–401.

[21] Langner, R. (2011). Stuxnet: Dissecting a cyberwarfare weapon. IEEE Security & Privacy, 9(3), 49–51.

[22] Lee, R. M., Assante, M. J., & Conway, T. (2016). Analysis of the cyber attack on the Ukrainian power grid [Report]. SANS Industrial Control Systems / Electricity Information Sharing and Analysis Center.

[23] Micali, S., Rabin, M., & Vadhan, S. (1999). Verifiable random functions. In Proceedings of the 40th Annual Symposium on Foundations of Computer Science (pp. 120–130). IEEE.

[24] Miller, A., Xia, Y., Croman, K., Shi, E., & Song, D. (2016). The honey badger of BFT protocols. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (pp. 31–42). ACM.

[25] Modbus Organization. (2012). Modbus application protocol specification V1.1b3.

[26] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf

[27] National Institute of Standards and Technology. (n.d.). Guide to industrial control systems (ICS) security (NIST Special Publication 800-82, Rev. 3).

[28] North American Electric Reliability Corporation. (n.d.). Critical infrastructure protection (CIP) reliability standards.

[29] Reyna, A., Martin, C., Chen, J., Soler, E., & Diaz, M. (2018). On blockchain and its integration with IoT: Challenges and

opportunities. Future Generation Computer Systems, 88, 173–190.

[30] SimPy Developers. (n.d.). SimPy: Discrete-event simulation for Python [Documentation]. https://simpy.readthedocs.io

[31] Yin, M., Malkhi, D., Reiter, M. K., Gueta, G. G., & Abraham, I. (2019). HotStuff: BFT consensus in the lens of blockchain. In Proceedings of the ACM Symposium on Principles of Distributed Computing (pp. 347–356). ACM.